Related Randomness Security for Public Key Encryption, Revisited

Takahiro Matsuda and Jacob C. N. Schuldt

National Institute of Advanced Industrial Science
and Technology (AIST), Tokyo, Japan
{t-matsuda,jacob.schuldt}@aist.go.jp

Abstract. Motivated by the history of randomness failures in practical systems, Paterson, Schuldt, and Sibborn (PKC 2014) introduced the notion of related randomness security for public key encryption. In this paper, we firstly show an inherent limitation of this notion: if the family of related randomness functions is sufficiently rich to express the encryption function of the considered scheme, then security cannot be achieved. This suggests that achieving security for function families capable of expressing more complex operations, such as those used in random number generation, might be difficult. The current constructions of related randomness secure encryption in the standard model furthermore reflect this; full security is only achieved for function families with a convenient algebraic structure. We additionally revisit the seemingly optimal random oracle model construction by Paterson et al. and highlight its limitations.

To overcome this difficulty, we propose a new notion which we denote *related refreshable randomness* security. This notion captures a scenario in which an adversary has limited time to attack a system before new entropy is added. More specifically, the number of encryption queries with related randomness the adversary can make before the randomness is refreshed, is bounded, but the adversary is allowed to make an unbounded total number of queries. Furthermore, the adversary is allowed to influence how entropy is added to the system. In this setting, we construct an encryption scheme which remains secure in the standard model for arbitrary function families of size 2^p (where p is polynomial in the security parameter) that satisfy certain collision-resistant and output-unpredictability properties. This captures a rich class of functions, which includes, as a special case, circuits of polynomial size. Our scheme makes use of a new construction of a (bounded) related-key attack secure pseudorandom function, which in turn is based on a new flavor of the leftover hash lemma. These technical results might be of independent interest.

1 Introduction

Most cryptographic primitives are designed under the assumption that perfect uniform randomness is available. However, in practice, this is often not the case. The design of random number generators (RNGs), which are used to generate

the required randomness, is a complex and difficult task, and several examples of RNGs failing in practice are known [23, 24, 26, 27, 20]. The consequences of this might be fatal, and the examples of attacks made possible by randomness failures are many (e.g. see [29, 32, 13, 12, 39]). To make matters worse, some cryptographic designs are particularly fragile with respect to randomness failures. An example of this, is the DSA signature scheme [33], which allows the signing key to be recovered from two signatures on different messages constructed using the same randomness. This property enabled the compromise of the security mechanisms in the Sony Playstation 3 [12], the theft of Bitcoins from wallets managed on Android devices [16], and the recovery of TLS server signing keys from virtualized servers [39]. The latter example highlights an important aspect: even if the used RNG is not flawed by itself, randomness failures might still occur when the RNG is used in virtualized environments which enable virtual machines (including the state of the RNG) to be cloned or reset. Given the risk of randomness failures occurring in practical systems, it is prudent to design cryptographic primitives that provide resilience against these to the extent that this is possible. While it is possible to address this via generic derandomization for primitives like signature schemes[1], this is not the case for other primitives like public key encryption, which inherently relies on randomness for security.

1.1 The Related Randomness Setting

Motivated by the challenge of designing public key encryption secure under randomness failure, Paterson, Schuldt, and Sibborn [34] introduced the notion of *related randomness attack* (RRA) security. This notion allows the adversary to control the randomness used in the encryption scheme, but still requires that messages encrypted under an honestly generated public key remain hidden, given that certain restrictions are placed on the adversary's queries. More specifically, the RRA security game defines a set of initially well-distributed random values which are hidden to the adversary. Via an encryption oracle, the adversary will be able to request encryptions under public keys and on messages of his choice, using functions ϕ of these random values. The adversary will furthermore have access to a challenge oracle, which, given two messages, consistently returns the encryption of the first or the second message under an honestly generated public key; the task of the adversary is to guess which of the messages is encrypted. However, even for the challenge encryptions, the adversary can specify functions ϕ of the random values defined in the game, which will be used as randomness in the encryptions. The RRA model is inspired by the practical attacks illustrated by Ristenpart and Yilek [39], which exploits weaknesses of randomness generation in virtual machines, and furthermore captures as a special case the reset attacks by Yilek [43] in which encryptions using repeated random values are considered.

[1] Specifically, the folklore approach of generating any required randomness via a keyed PRF evaluated on the message to be signed, will work for any signature scheme. See also discussion of deterministic DSA in [36].

In [34], Paterson *et al.* showed several constructions of schemes secure in the RRA setting. Specifically, assuming the functions ϕ are drawn from a function family Φ of *output-unpredictable* and *collision-resistant* functions (which are also necessary conditions for achieving RRA security), the simple randomized-encrypt-with-hash (REwH) scheme by Bellare *et al.* [6] is shown to achieve RRA security in the random oracle model (however, as will be explained below, this construction still suffers from limitations inherent to the RRA model). Furthermore, in the standard model, a generic construction based on a Φ-related key attack secure pseudo-random function (RKA-PRF) [7] and any standard encryption scheme, is shown to yield a RRA-secure encryption for functions Φ. Using recent constructions of RKA-PRFs, e.g. [3], an encryption scheme RRA-secure for polynomial functions Φ can be obtained. Likewise, a generic construction based on a Φ-correlated input-secure (CIS) hash function [25], a standard PRF, and an encryption scheme, is shown to yield a RRA-secure encryption scheme for functions Φ, albeit in a weaker *honest-key* model. Furthermore, the only known standard model construction of a CIS hash function only provides selective security for polynomial functions Φ. In more recent work, Paterson et al. [35] showed a generic construction based on a reconstructive extractor and an encryption scheme, which yields security for *hard-to-invert* function families, but only in a selective security model in which the adversary is forced to commit to the functions used in the security game before seeing the public key. Furthermore, the concrete construction obtained in [35] only allows the adversary to maliciously modify the randomness used by his encryption oracle; the challenge oracle is required to use uniformly distributed randomness.

Hence, the best known construction achieving a reasonable level of security in the standard model, only obtains RRA-security for polynomial function families Φ. However, it seems unlikely that the randomness relations encountered in practice can be expressed with a function class with such convenient algebraic structure. While obtaining security for more complex function classes is clearly desirable, it is challenging to construct provably secure schemes for function families without an algebraic structure that can be exploited in the proof. This challenge is additionally reflected by the current state-of-the-art RKA-secure PRFs [3, 1] which can only handle polynomial function families.

1.2 Our Results

First of all, we observe that if the function family Φ becomes sufficiently complex, RRA-security cannot be achieved for Φ. More precisely, if Φ is sufficiently rich to be able to express the encryption function of the scheme we are considering, a direct attack against the scheme in the RRA setting becomes possible. The attack is relatively simple, and is based on the ability of the adversary to derive the randomness used in his challenge encryption with the help of his encryption oracle. Assuming the encryption scheme satisfies ordinary IND-CPA security, the attack does not violate the properties required to make the RRA-security notion meaningful, which are the equality-respecting property, output unpredictability, and collision resistance. The details of this are given in Section 4. At first, this

might appear to contradict the results by Paterson *et al.* [34] regarding the REwH construction in the random oracle model. However, closer inspection reveals that the results from [34] implicitly assume that the functions Φ are *independent* of the random oracle, and hence, Φ will not be able to capture the encryption function of the REwH construction.

Considering the above, we revisit the security of the REwH construction in the random oracle model, and show that if additional restrictions are placed on the adversary, security can be obtained. More specifically, if the adversary respects *indirect H-query uniqueness*, which is a property requiring that the random oracle queries caused by the adversary's encryption and challenge queries are all distinct, then RRA-security is obtained, even for function families Φ which are dependent on the random oracle, as long as the functions in Φ are output-unpredictable. The details of this are in Section 5. Our results are reminiscent of the results by Albrecht et al. [5] regarding cipher-dependent related-key attacks in the ideal cipher model.

However, the indirect H-query uniqueness property is an artificial restriction to place on the adversary, and the above result seems unsatisfactory. Furthermore, the above negative result suggests that, achieving security for function families that reflect more complex operations, which might be used in random number generators, could be difficult.

Hence, to overcome this difficulty, we propose a new notion which we denote *related refreshable randomness* security. In this notion, we bound the number of queries an adversary can make before new entropy is added to the system, but allow an unbounded total number of queries. We refer to the periods between refreshes as *epochs*. Furthermore, we allow the adversary to maliciously influence how entropy is added between epochs. This is implemented by giving the adversary access to a *refresh* oracle through which the adversary can submit update functions ψ. These functions take as input the current random values and a update seed chosen uniformly at random, and output new random values which will be used in the security game. For this update mechanism to be meaningful, we restrict the functions ψ to come from a function family Ψ in which all functions have the property, that their output has a certain level of min-entropy conditioned on the random values being updated (i.e. it is required that a certain amount of the entropy contained in the update seed, will be carried over to the output of the update function). With this requirement in place, we consider adversaries who makes at most n queries to their encryption and challenge oracles, before querying the refresh oracle. The details of the security model are given in Section 3.

The related refreshable randomness setting models the arguably realistic scenario in which an attacker only has limited time to interact with a system that is in a state where no new entropy is being added to the system, and highly correlated randomness values are used for encryption. This furthermore resembles the observations made in [39] regarding virtual machine reset attacks; the attacks were only possible in a relatively short window after the virtual ma-

chine was reset, before sufficient entropy was gathered from the network, clock synchronization, and similar sources.

The related refreshable randomness setting furthermore allows us to obtain positive results in the standard model. Specifically, we construct a scheme which is secure in the related refreshable randomness setting for *arbitrary* function families Φ and Ψ satisfying certain output unpredictability and collision resistance properties. We do, however, require the size of the function families to be bounded by an a priori known bound of the form 2^p, where p is a polynomial in the security parameter. This allows us to capture a rich class of functions which include, for example, the set of all functions that can be described by circuits of polynomial size. Our construction is based on the same high-level approach as taken in [34] and [43], and combines a standard encryption scheme with a PRF (see below for the details). However, by relying on a new construction of a (bounded) RKA-secure PRF, we are able to prove security in the related refreshable randomness setting for much more interesting function classes than considered in [34] and [43]. Notably, in contrast to our scheme, the scheme from [43] is only reset secure ($\Phi = \{\mathtt{id}\}$), and the scheme from [34] only achieves selective security for polynomial functions Φ, and hence cannot capture non-algebraic functions such as bit-flipping and bit-fixing, which are highly relevant to randomness failures in practice. The full details can be found in Section 7.

1.3 Technique

As highlighted above, the main tool we use to obtain our standard model encryption scheme secure in the related refreshable randomness setting, is a new construction of a RKA-secure PRF. We consider this construction to be our main technical contribution. As an intermediate step, we construct (a variant of) a CIS hash function. This type of hash function was originally introduced by Goyal, O'Neill, and Rao [25]. While different security notions for CIS hash functions were introduced in [25], the one we will be concerned with here, is pseudo-randomness. This notion requires that, for a hash function $H : D \to R$ and a randomly chosen value $x \in D$, an adversary cannot distinguish an oracle which returns $H(\phi(x))$ for adversarially chosen functions ϕ, from an oracle that returns a random value from R. In [25], a construction obtaining selective security for a polynomial function family Φ was shown. However, we show that by bounding the number of queries to the adversary's oracle, we can obtain a construction achieving security for a class Φ of arbitrary functions that are output-unpredictable and collision-resistant, where the size of Φ is bounded a priori. This construction is in turn based on a new flavor of the leftover hash lemma [28] for correlated inputs that might depend on the description of the hash function. Then, by applying this CIS hash function H to the key of a standard PRF $\mathtt{prf}$, we obtain a new PRF $\mathtt{prf}'(k, x) := \mathtt{prf}(H(k), x)$ that provides RKA security, as long as the adversary will only query a bounded number of different key derivation functions. However, the adversary is allowed to obtain an unbounded number of evaluation results under the derived keys. The detailed proofs of security can be found in Section 6.

Finally, we obtain a standard model encryption scheme in the related refreshable randomness setting via the same transformation used in [43] and [34]: to encrypt a message m under public key pk using randomness r, we compute $\mathtt{Enc}(pk, m; r')$, where $r' = \mathtt{prf}'(r, pk \| m)$. The security properties of the constructed PRF $\mathtt{prf}'$ allows us to prove security via a hybrid argument with respect to the epochs. Note, however, that the parameters of the scheme will grow linearly in the in the number of queries an adversary is allowed to make in each epoch, as a description of H must be included. See Section 7 for the details.

Our construction of a RKA-secure PRF, CIS hash function, and our new flavor of the leftover hash lemma, might find applications outside of related randomness security, and hence, might be of independent interest. For example, by directly applying our RKA-secure PRF in combination with the framework of Bellare, Cash, and Miller [8], we can obtain RKA-secure signatures, public key encryption, and identity-based encryption for function families of size bounded by 2^p and with the appropriate collision-resistant and output-unpredictability properties. Security is only guaranteed for a bounded number of related key derivation queries, but the total number of allowed signatures, decryption queries, and key queries for identities, respectively, is unbounded. Furthermore, it is not hard to see that our PRF construction only requires the PRF keys to have high min-entropy (as opposed to being uniformly distributed), as long as the considered function family remains collision-resistant and output-unpredictable. This indicates that the construction can additionally tolerate leakage, and we conjecture that bounded leakage and tampering security as defined by Damgård et al. [18, 19], can be achieved.

1.4 Related work

A number of works in the literature have considered security of various cryptographic primitives in the event of randomness failures. In the symmetric key setting, Rogaway and Shrimpton [40] considered the security of authenticated encryption in the case nonces are repeated, and Katz and Kamara [31] considered chosen randomness attacks which allows the adversary to freely choose the randomness, except for the challenge encryption. In the public key setting, Bellare *et al.* [6] considered *hedged* encryption, which remains secure as long as the joint distribution of messages and randomness contains sufficient entropy. Note that the security notion formalized for hedged encryption in [6], security against chosen distribution attacks (CDA), is incomparable to RRA-security which does not rely on message entropy. Furthermore, whereas RRA-security allows the adversary to obtain encryptions under maliciously chosen public keys using randomness related to the randomness of the challenge encryptions, there is no equivalent in CDA-security, and CDA-security does not allow messages and randomness to depend on the public key. Additionally, the known standard model constructions of CDA-secure schemes are only shown secure for block sources which require each message/randomness pair to have high min-entropy conditioned on all previous pairs, whereas the standard model RRA-secure schemes from [34, 35] and the schemes in this paper do not have similar restrictions. Vergnaud

and Xaio [42] slightly strengthened the CDA-security considered in [6] by allowing the message/randomness pair to partly depend on the public key. Yilek [43] considered reset attacks in which encryptions with repeated randomness values might occur, and gave a construction based on a standard encryption scheme and a PRF. This is a special case of the RRA-setting. Bellare and Tackmann [11] introduced the notion of nonce-based public key encryption, and achieved a number of strong results. However, the constructions assume a stateful scheme, and is hence not applicable to a number of scenarios in which we are interested in related randomness security, e.g. virtual machine resets. Extending [6] and [11], Hoang et al. [30] considered security of hedged encryption and nonce-based public key encryption under selective opening attack.

Appelbaum and Widder [4] constructed a (bounded) RKA-secure PRF for additions, while Abdalla *et al.* [2] constructed a RKA-secure PRF for XORs from multilinear maps. In contrast, our PRF construction achieves security for arbitrary functions satisfying collision resistance and unpredictability, for a bounded number of related keys. We stress, however, that the bound is only on the number of keys, and that our construction remains secure for an unbounded number of PRF evaluations.

2 Preliminaries

2.1 Notation and Basic Notions

Throughout the paper we will use $\lambda \in \mathbb{N}$ to denote the security parameter, which will sometimes be written in its unary representation, 1^λ. Furthermore, we sometimes suppress the dependency on λ, when λ is clear from the context. We denote by $y \leftarrow x$ the assignment of y to x, and by $s \leftarrow_\$ S$ we denote the selection of an element s uniformly at random from the set S. The notation $[n]$ represents the set $\{1, 2, \ldots, n\}$. For an algorithm A, we denote by $y \leftarrow A(x; r)$ that A is run with input x and random coins r, and that the output is assigned to y. For a vector $\boldsymbol{x} = (x_1, x_2, \ldots)$, we denote by $A(\boldsymbol{x})$ the vector $(A(x_1), A(x_2), \ldots)$. For a random variable X defined over a set S, we denote by $\mathtt{H}_\infty(X)$ the min-entropy of X (i.e. $\mathtt{H}_\infty(X) = -\log_2 \max_{x \in S} \Pr[X = x]$), and for two random variables X and Y defined over the same set S, we denote the statistical distance between X and Y as $\Delta[X, Y]$ (i.e. $\Delta[X, Y] = \frac{1}{2} \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]|$).

2.2 t-wise Independent Hash Functions

One of the basic building blocks of our construction is t-wise independent hash functions, which we define here. We furthermore recall a tail inequality for t-wise independent variables due to Bellare and Rompel [10], which we will make use of in our proofs of security.

Definition 1 (t-wise independent hash function family) *Let $\mathcal{H} = \{H \mid H : D \to R\}$ be a family of hash functions. $\mathcal{H}$ is said to be a* t-wise independent hash function family*, if for all mutually distinct $x_1, \ldots, x_t \in D$ and all $y_1, \ldots, y_t \in R$, it holds that $\Pr_{H \leftarrow_\$ \mathcal{H}}[H(x_1) = y_1 \wedge \cdots \wedge H(x_t) = y_t] = \frac{1}{|R|^t}$.*

Theorem 1 (Tail inequality [10]) *Let t be an even integer larger than 8, and let $X_1, \ldots, X_n$ be t-wise independent variables*[2] *assuming values in the interval $[0,1]$. Furthermore, let $X = X_1 + \ldots + X_n$, $\mu = \mathtt{E}[X]$, and $\epsilon < 1$. Then*

$$\Pr[|X - \mu| \geq \epsilon\mu] \leq \left(\frac{t}{\epsilon^2\mu}\right)^{t/2}.$$

2.3 Output Unpredictability and Collision Resistance

We will consider function families which are output-unpredictable and collision-resistant. These properties were originally defined by Bellare et al. [9] in the context of RKA security, and used by Paterson et al. [34] in the context of RRA security. The following definitions are slightly simplified compared to [9, 34].

Definition 2 (Output unpredictability) *Let $\Phi = \{\phi : D \to R\}$ be a family of functions with domain $D = D_\lambda$ and range $R = R_\lambda$. The output unpredictability of Φ is defined as $\mathsf{UP}^{\Phi}(\lambda) = \max_{\phi \in \Phi, y \in R} \Pr[x \leftarrow_\$ D : \phi(x) = y]$. When $\mathsf{UP}^{\Phi}(\lambda) < \epsilon$ for a negligible function $\epsilon = \epsilon(\lambda)$, we simply say that Φ is* output-unpredictable*.*

Definition 3 (Collision resistance) *Let $\Phi = \{\phi : D \to R\}$ be a family of functions with domain $D = D_\lambda$ and range $R = R_\lambda$. The collision resistance of Φ is defined as $\mathsf{CR}^{\Phi}(\lambda) = \max_{\phi_1, \phi_2 \in \Phi, \phi_1 \neq \phi_2} \Pr[x \leftarrow_\$ D : \phi_1(x) = \phi_2(x)]$. When $\mathsf{CR}^{\Phi}(\lambda) < \epsilon$ for a negligible function $\epsilon = \epsilon(\lambda)$, we simply say that Φ is* collision-resistant*.*

2.4 Pseudorandom Function

A pseudorandom function $\mathtt{F}$ is given by the following three algorithms: $\mathtt{F.Setup}(1^\lambda)$ which on input the security parameter, returns public parameters *par* (required to describe a domain D and a range R); $\mathtt{F.KeyGen}(par)$ which, on input *par*, returns a key k; and $\mathtt{F.Eval}(par, k, x)$ which, on input *par*, key k, and input $x \in D$, returns an output value $y \in R$. For notational convenience, we will sometimes suppress *par* from the input.

We will consider the security of a pseudorandom function in a multi-key setting. This is for convenience only; by a standard hybrid argument, it is easily seen that this definition is equivalent to a definition considering a single key, as also shown by Bellare et al. [15]. We define security via the security game shown in Figure 1.

Definition 4 *Let the advantage of an adversary $\mathcal{A}$ playing the security game in Figure 1 with respect to a pseudorandom function $\mathtt{F} = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Eval})$ be defined as $\mathtt{Adv}^{\mathtt{PRF}}_{\mathtt{F},\mathcal{A}}(\lambda) = 2\left|\Pr[\mathrm{PRF}^{\mathtt{F}}_{\mathcal{A}}(\lambda) \Rightarrow 1] - \frac{1}{2}\right|$. $\mathtt{F}$ is said to be secure if for all PPT adversaries $\mathcal{A}$, $\mathtt{Adv}^{\mathtt{PRF}}_{\mathtt{F},\mathcal{A}}(\lambda)$ is negligible in the security parameter λ.*

[2] Random variables $X_1, \ldots, X_n$ are *t-wise independent* if for all distinct indices $i_1, \ldots, i_t \in [n]$ and all $x_1, \ldots, x_t$, $\Pr[\bigwedge_{j \in [t]}(X_{i_j} = x_j)] = \prod_{j \in [t]} \Pr[X_{i_j} = x_j]$ holds.

$\underline{\mathrm{PRF}^{\mathtt{F}}_{\mathcal{A}}(\lambda)}$:
$par \leftarrow \mathtt{F.Setup}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$
$\mathcal{F} \leftarrow \emptyset$
$ctr \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\textsc{Eval},\textsc{New}}(par)$
return $(b = b')$

proc. $\textsc{Eval}(i, x)$:
if $i > ctr$, return $\perp$
if $b = 1$
  $y \leftarrow \mathtt{F.Eval}(k_i, x)$
else
  if $\mathcal{F}[i, x] = \perp$, $\mathcal{F}[i, x] \leftarrow_\$ R$
  $y \leftarrow \mathcal{F}[i, x]$
return y

proc. $\textsc{New}$:
$ctr \leftarrow ctr + 1$
$k_{ctr} \leftarrow \mathtt{F.KeyGen}(par)$
return ctr

Fig. 1. Game defining security of a pseudorandom function.

$\underline{\text{IND-CCA}^{\mathtt{PKE}}_{\mathcal{A}}(\lambda)}$:
$par \leftarrow \mathtt{PKE.Setup}(1^\lambda)$
$(pk^*, sk^*) \leftarrow \mathtt{PKE.KeyGen}(par)$
$b \leftarrow_\$ \{0,1\}$
$\mathcal{C} \leftarrow \emptyset$
$b' \leftarrow \mathcal{A}^{\text{LR},\textsc{Dec}}(par, pk^*)$
return $(b = b')$

proc. $\text{LR}(m_0, m_1)$:
$c \leftarrow \mathtt{PKE.Enc}(pk^*, m_b)$
$\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$
return c

proc. $\textsc{Dec}(c)$:
if $c \in \mathcal{C}$,
  return $\perp$
return $\mathtt{PKE.Dec}(sk^*, c)$

Fig. 2. Game defining `IND-CCA` security for a PKE scheme.

2.5 Public Key Encryption

A public key encryption (PKE) scheme `PKE` is defined by the following four algorithms: $\mathtt{PKE.Setup}(1^\lambda)$ which on input the security parameter, returns public parameters par; $\mathtt{PKE.KeyGen}(par)$ which on input par, returns a public/private key pair (pk, sk); $\mathtt{PKE.Enc}(par, pk, m)$ which on input par, public key pk, and message m, returns an encryption c of m under pk; and $\mathtt{PKE.Dec}(par, sk, c)$ which on input par, private key sk, and ciphertext c, returns either a message m or the error symbol $\perp$. For convenience, we often suppress par from the input.

We require that a PKE scheme satisfies *perfect correctness*, that is, for all λ, all $par \leftarrow \mathtt{PKE.Setup}(1^\lambda)$, all $(pk, sk) \leftarrow \mathtt{PKE.KeyGen}(par)$, and all $m \in \mathcal{M}(pk)$, it holds that $\mathtt{PKE.Dec}(sk, \mathtt{PKE.Enc}(pk, m)) = m$. Security of a PKE scheme is defined via the game shown in Figure 2.

Definition 5 (`IND-CCA` security) *Let the advantage of an adversary $\mathcal{A}$ playing the* `IND-CCA` *game with respect to a PKE scheme* $\mathtt{PKE} = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$*, be defined as:* $\mathtt{Adv}^{\mathtt{IND\text{-}CCA}}_{\mathtt{PKE},\mathcal{A}}(\lambda) = 2\left|\Pr[\text{IND-CCA}^{\mathtt{PKE}}_{\mathcal{A}}(\lambda) \Rightarrow 1] - \frac{1}{2}\right|$*. A scheme* `PKE` *is said to be* `IND-CCA` *secure, if for all PPT adversaries $\mathcal{A}$,* $\mathtt{Adv}^{\mathtt{IND\text{-}CCA}}_{\mathtt{PKE},\mathcal{A}}(\lambda)$ *is negligible in the security parameter λ.*

3 Related Refreshable Randomness Security

We will firstly define our new notion of related refreshable randomness security. This builds upon the RRA-security notion defined by Paterson *et al.* [34], but models a setting in which the adversary has limited time to attack a system before new entropy is added to the system. As in the original RRA security game,

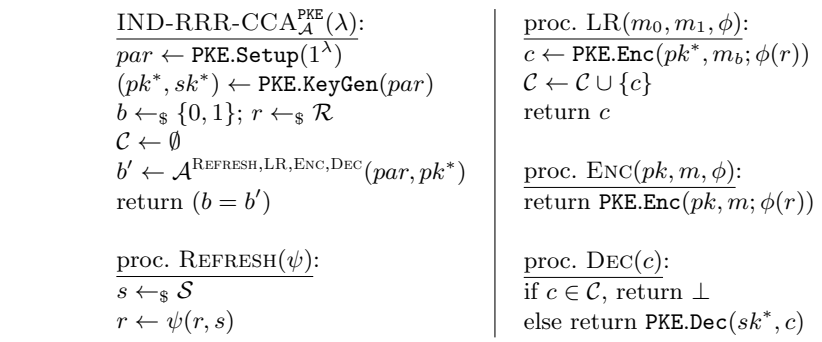

Fig. 3. Game defining indistinguishability under related refreshable randomness and chosen ciphertext attacks (`IND-RRR-CCA`).

we consider a polynomial number of randomness values r_i, and give the adversary access to an encryption oracle Enc which returns encryptions under public keys and messages of the adversary's choice, and a challenge left-or-right oracle LR, which consistently returns the encryption of either the first or the second message of two submitted messages m_0, m_1, under an honestly generated challenge public key pk^*. However, for both oracles, the adversary can not only specify which random value r_i to be used, but also a function ϕ which will be applied to r_i before it is used (i.e. the used randomness will be $\phi(r_i)$). We furthermore introduce an additional oracle, Refresh, which allows the adversary to submit a function ψ that will be used to refresh the random values r_i. The function ψ takes two inputs: the randomness r_i which is to be refreshed, and a seed s. Here, the seed s will be drawn uniformly at random from a seed space $\mathcal{S}$, and $\psi : \mathcal{R} \times \mathcal{S} \to \mathcal{R}$, where $\mathcal{R}$ is the randomness space of the encryption scheme. The full security game is defined in Figure 3. Note that while the security game shown in Figure 3 is only defined for a single random value r, this is equivalent to a model defined for a polynomial number of randomness values r_i (see the full version of the paper).

Note that, by itself, introducing the Refresh oracle does not achieve the intended goal, as the adversary is not forced to query Refresh. However, we will consider a class of adversaries which make at most n Enc and LR queries between each call to Refresh (but is allowed to make an unrestricted number of queries to Dec). We will furthermore parameterize this class of adversaries by function families Φ and Ψ from which an adversary is allowed to choose related randomness functions ϕ and refresh functions ψ, respectively, and will refer to adversaries in this class as (n, Φ, Ψ)-restricted adversaries[3]. In the following def-

[3] Note that since the functions ϕ and ψ will depend on the security parameter λ, Φ and Ψ are technically ensembles of function families indexed by λ. However, for notational convenience, we suppress λ.

initions and proofs, we need to refer to the execution of an adversary in between two calls to REFRESH, which we will denote an *epoch*[4].

As in the case of RRA-security, since the defined oracles let the adversary control the randomness in the challenge encryptions, a few natural restrictions must be placed on the adversary's queries to obtain a meaningful definition of security. Specifically, we require that an adversary is *equality respecting*. This is reminiscent of the restriction defined for deterministic encryption schemes [38].

Definition 6 (Equality-respecting adversary) *Consider a (n, Φ, Ψ)-restricted adversary $\mathcal{A}$ playing the* IND-RRR-CCA *security game for security parameter λ. Let $\mathcal{M}_{\mathtt{Enc}}^{\phi,\delta}$ denote the set of messages $\mathcal{A}$ submits to the* ENC *oracle for challenge public key pk^* and related randomness function $\phi \in \Phi$ in refresh epoch δ. Furthermore, let $(m_0^{\phi,\delta,1}, m_1^{\phi,\delta,1}), \ldots, (m_0^{\phi,\delta,q_\phi}, m_1^{\phi,\delta,q_\phi})$ denote the messages $\mathcal{A}$ submits to the* LR *oracle for function ϕ in refresh epoch δ. Then $\mathcal{A}$ is said to be* equality-respecting *if, for all $\phi \in \Phi$, for all refresh epochs δ, and for all $i, j \in [q_\phi]$ s.t. $i \neq j$,*

$$m_0^{\phi,\delta,i} = m_0^{\phi,\delta,j} \Leftrightarrow m_1^{\phi,\delta,i} = m_1^{\phi,\delta,j} \quad \textit{and} \quad m_0^{\phi,\delta,i}, m_1^{\phi,\delta,j} \notin \mathcal{M}_{\mathtt{Enc}}^{\phi,\delta}.$$

With this definition in place, we are ready to define our notion of security.

Definition 7 (IND-RRR-CCA Security) *Let the advantage of an adversary $\mathcal{A}$ playing the* IND-RRR-CCA *game with respect to a public key encryption scheme* PKE = (PKE.Setup, PKE.KeyGen, PKE.Enc, PKE.Dec)*, be defined as:*

$$\mathtt{Adv}_{\mathtt{PKE},\mathcal{A}}^{\mathtt{IND\text{-}RRR\text{-}CCA}}(\lambda) = 2\left|\Pr[\text{IND-RRR-CCA}_{\mathtt{PKE}}^{\mathcal{A}}(\lambda) \Rightarrow 1] - \frac{1}{2}\right|.$$

A scheme PKE *is said to be (n, Ψ, Φ)-*IND-RRR-CCA *secure, if for all PPT (n, Φ, Ψ)-restricted and equality-respecting adversaries $\mathcal{A}$, $\mathtt{Adv}_{\mathtt{PKE},\mathcal{A}}^{\mathtt{IND\text{-}RRR\text{-}CCA}}(\lambda)$ is negligible in the security parameter λ.*

The original RRA-security notion defined in [34] can be obtained from the above definition by not allowing the adversary access to the REFRESH oracle (i.e. considering only the first refresh epoch) and considering an unbounded value n. In this case, Ψ is irrelevant, and we simply write Φ-IND-RR-CCA security to denote this security notion[5]. Lastly, note that ordinary IND-CCA security can be obtained from the above definition by setting $n = 1$, $\Phi = \{\mathtt{id}\}$, and $\Psi = \{\mathtt{id}_2 : (r, s) \to s\}$ (assuming $\mathcal{S} = \mathcal{R}$).

3.1 Basic Function Family Restrictions

Unsurprisingly, related randomness security for all function families Φ and Ψ is not achievable. This is similar to the security notions for related key attacks (e.g.

[4] Hence, if an adversary $\mathcal{A}$ submits q queries to REFRESH in total, then the execution of $\mathcal{A}$ has $q + 1$ epochs.

[5] Note that in [34], the notation Φ-RRA-CCA was used for this security notion.

see [7]), which must restrict the class of related-key deriving functions that can be applied to the private key, in order to become achievable. We will now establish basic restriction which must be placed on Φ and Ψ to make the IND-RRR-CCA notion defined above achievable.

The two basic properties we consider are output-unpredictability and collision-resistance of the functions in Φ. However, as the IND-RRR-CCA security game allows the adversary to update the challenge randomness using the functions Ψ, we will consider output-unpredictability and collision-resistance of Φ with respect to Ψ i.e. the functions in Φ must be output-unpredictable and collision-resistant, even when the input is modified using functions from Ψ. In the following definitions we will use the notation $\overline{\Psi}^q$ to denote the *q-closure* of the functions in Ψ. More specifically, each function $\overline{\psi} \in \overline{\Psi}^q$ corresponds to q updates of a randomness value r using q functions $\psi_1, \ldots, \psi_q \in \Psi$, and will take as input r and q seeds $\overline{s} = (s_1, \ldots, s_q)$ and return $\overline{\psi}(r, \overline{s}) = \psi_q(\psi_{q-1}(\cdots \psi_1(r, s_1) \cdots, s_{q-1}), s_q)$. As the seeds s_i are elements of $\mathcal{S}$, we have that $\overline{\psi} : \mathcal{R} \times \mathcal{S}^q \to \mathcal{R}$.

Definition 8 (Output-unpredictability of Φ w.r.t. Ψ) *Let $\Phi = \{\phi : \mathcal{R} \to \mathcal{R}\}$ and $\Psi = \{\psi : \mathcal{R} \times \mathcal{S} \to \mathcal{R}\}$ be function families, where $\mathcal{R} = \mathcal{R}_\lambda$ and $\mathcal{S} = \mathcal{S}_\lambda$. For a positive integer q, the q-output-unpredictability of Φ with respect to Ψ is defined as* $\mathsf{UP}_q^{\Phi,\Psi}(\lambda) = \max_{\phi \in \Phi, \overline{\psi} \in \overline{\Psi}^q, y \in \mathcal{R}} \Pr\left[r \leftarrow_\$ \mathcal{R}, \overline{s} \leftarrow_\$ \mathcal{S}^q : \phi(\overline{\psi}(r, \overline{s})) = y\right]$.

Definition 9 (Collision-resistance of Φ w.r.t. Ψ) *Let $\Phi = \{\phi : \mathcal{R} \to \mathcal{R}\}$ and $\Psi = \{\psi : \mathcal{R} \times \mathcal{S} \to \mathcal{R}\}$ be function families, where $\mathcal{R} = \mathcal{R}_\lambda$ and $\mathcal{S} = \mathcal{S}_\lambda$. The collision-resistance of Φ with respect to Ψ is defined as*

$$\mathsf{CR}_q^{\Phi,\Psi}(\lambda) = \max_{\substack{\phi_1, \phi_2 \in \Phi, \overline{\psi} \in \overline{\Psi}^q \\ \phi_1 \neq \phi_2}} \Pr\left[r \leftarrow_\$ \mathcal{R}, \overline{s} \leftarrow_\$ \mathcal{S}^q : \phi_1(\overline{\psi}(r, \overline{s})) = \phi_2(\overline{\psi}(r, \overline{s}))\right].$$

In [34], Paterson *et al.* showed that to achieve Φ-IND-RR-CCA security, Φ is required to satisfy standard output-unpredictability and collision-resistance. Likewise, in the IND-RRR-CCA setting, we can show that Φ must be output-unpredictability and collision-resistance w.r.t. Ψ for security to be achievable.

Theorem 2 (Necessity of Φ output-unpredictability w.r.t. Ψ) *Let $\Psi = \{\psi : \mathcal{R} \times \mathcal{S} \to \mathcal{R}\}$ be a function family, where $\mathcal{R} = \mathcal{R}_\lambda$ and $\mathcal{S} = \mathcal{S}_\lambda$, and suppose that there exist a positive integer $q = \mathsf{poly}(\lambda)$ and a non-negligible function $\epsilon = \epsilon(\lambda)$ such that $\mathsf{UP}_q^{\Phi,\Psi}(\lambda) > \epsilon$. Then no PKE scheme can be (n, Ψ, Φ)-IND-RRR-CCA secure for $n \geq 1$.*

Proof (Sketch). The proof is straightforward. Let $\phi \in \Phi$, $\overline{\psi} \in \overline{\Psi}^q$, and $y \in \mathcal{R}$ such that $\Pr[r \leftarrow_\$ \mathcal{R}, \overline{s} \leftarrow_\$ \mathcal{S}^q : \phi(\overline{\psi}(r, \overline{s})) = y] > \epsilon$. These are guaranteed to exist since $\mathsf{UP}_q^{\Phi,\Psi}(\lambda) > \epsilon$. Consider an adversary $\mathcal{A}$ submitting functions corresponding to $\overline{\psi}$ as REFRESH queries, and (ϕ, m_0, m_1) in a following LR query. Let c be the challenge ciphertext $\mathcal{A}$ receives. Now, let $\mathcal{A}$ check whether $c = \mathtt{Enc}(pk^*, m_b; y)$ for $b = 0$ and $b = 1$, and if so, return b. Otherwise, let $\mathcal{A}$ return a random bit. It easily follows that such $\mathcal{A}$ has advantage at least ϵ which is assumed to be non-negligible, and hence the considered PKE scheme cannot be secure. □ (**Theorem 2**)

Theorem 3 (Necessity of Φ collision-resistance w.r.t. Ψ) *Let $\Phi = \{\phi : \mathcal{R} \to \mathcal{R}\}$ and $\Psi = \{\psi : \mathcal{R} \times \mathcal{S} \to \mathcal{R}\}$ be function families, where $\mathcal{R} = \mathcal{R}_\lambda$ and $\mathcal{S} = \mathcal{S}_\lambda$. Suppose that there exist a positive integer $q = \mathsf{poly}(\lambda)$ and a non-negligible function $\epsilon = \epsilon(\lambda)$ such that $\mathsf{CR}_q^{\Phi,\Psi}(\lambda) > \epsilon$. Then no PKE scheme can be (n, Ψ, Φ)-IND-RRR-CCA secure for $n \geq 2$.*

The proof of this theorem is similar to the proof of Theorem 2 and is omitted.

Note that, without further assumptions on Ψ, queries to the REFRESH oracle is not guaranteed to change the random value r used to respond to ENC and LR queries. In particular, if $\Psi = \{\mathtt{id}_1 : (r, s) \to r\}$, the original value of r will be used in every refresh epoch, which essentially corresponds to removing the bound n on the number of ENC and LR queries. However, it is relatively easy to see that security cannot be achieved in this case[6]. Furthermore, the very idea behind introducing the IND-RRR-CCA security notion is to show that a guarantee of new entropy is being added to the system with certain intervals, can be leveraged to provide stronger security properties. Hence, we will consider a function class Ψ for which the output $r' \leftarrow \psi(r, s)$ of all update functions $\psi \in \Psi$ is required to depend on the seed s, or more specifically, that $\psi(r, s)$ will have a certain level of conditional min-entropy given r. We introduce this requirement implicitly via the following slightly stronger notions of output-unpredictability and collision-resistance of Φ w.r.t. Ψ. These notions require that the functions in Φ remain output-unpredictable and collision-resistant on input $\psi(r', s)$, $\psi \in \Psi$, for a randomly chosen seed s and any value r', as opposed to a value of r' obtained by choosing the initial r at random and then modifying this using a chain of update functions $\overline{\psi} \in \overline{\Psi}^q$ and corresponding seeds $\overline{s} \in \mathcal{S}^q$. We refer to these notions as *seed-induced* output-unpredictability and collision-resistance.

Definition 10 (Seed-induced output-unpredictability of Φ w.r.t. Ψ) *Let $\Phi = \{\phi : \mathcal{R} \to \mathcal{R}\}$ and $\Psi = \{\psi : \mathcal{R} \times \mathcal{S} \to \mathcal{R}\}$ be function families, where $\mathcal{R} = \mathcal{R}_\lambda$ and $\mathcal{S} = \mathcal{S}_\lambda$. The seed-induced output-unpredictability of Φ with respect to Ψ is defined as*

$$\mathsf{sUP}^{\Phi,\Psi}(\lambda) = \max_{\phi \in \Phi, \psi \in \Psi, r, y \in \mathcal{R}} \Pr\left[s \leftarrow_\$ \mathcal{S} : \phi(\psi(r, s)) = y\right].$$

Definition 11 (Seed-induced collision-resistance of Φ w.r.t. Ψ) *Let $\Phi = \{\phi : \mathcal{R} \to \mathcal{R}\}$ and $\Psi = \{\psi : \mathcal{R} \times \mathcal{S} \to \mathcal{R}\}$ be function families, where $\mathcal{R} = \mathcal{R}_\lambda$ and $\mathcal{S} = \mathcal{S}_\lambda$. The seed-induced collision-resistance of Φ with respect to Ψ is defined as*

$$\mathsf{sCR}^{\Phi,\Psi}(\lambda) = \max_{\substack{\phi_1, \phi_2 \in \Phi, \psi \in \Psi, r \in \mathcal{R} \\ \phi_1 \neq \phi_2}} \Pr\left[s \leftarrow_\$ \mathcal{S} : \phi_1(\psi(r, s)) = \phi_2(\psi(r, s))\right].$$

[6] In particular, the above definition of an equality-respecting adversary will allow the messages m_0, m_1 and the function ϕ from a LR query in one refresh epoch, to be submitted to the ENC oracle in combination with pk^* in a different refresh epoch, which trivially allows the adversary to break security.

4 Restrictions on the Complexity of Function Families

We will now turn our attention to function families which satisfy the basic output-unpredictability and collision-resistant properties, but for which security nevertheless cannot be achieved.

More specifically, when Φ and Ψ become rich enough to express the encryption function itself of a scheme, a direct attack against the scheme becomes possible. This is reminiscent of the results by Albrecht et al. [5] regarding cipher-dependent related-key attacks in the ideal cipher model. The attack is based on the ability of an adversary to force the challenge encryption to be constructed using a value which can be obtain through the ENC and LR oracles available to the adversary. This is captured by the following theorem.

Theorem 4 *Let* $\mathtt{PKE} = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ *be a public key encryption scheme, and let* Φ *be a family of functions such that* $\mathtt{id} \in \Phi$ *and* $f(\mathtt{Enc}(pk, m, \cdot)) \in \Phi$ *for some public key* pk*, message* m*, and a mapping function* $f : \mathcal{C} \to \mathcal{R}$*, where* $\mathcal{C}$ *and* $\mathcal{R}$ *are the ciphertext space and randomness space of* PKE*, respectively. Then* PKE *cannot be* (n, Ψ, Φ)-IND-RRR-CCA *secure for any* $n \geq 2$ *and any function family* Ψ*.*

Proof. The proof is straightforward. Since it is assumed that $f(\mathtt{Enc}(pk, m; \cdot)) \in \Phi_\lambda$, an adversary would be able to submit $\phi(\cdot) = f(\mathtt{Enc}(pk, m; \cdot))$ and two distinct messages, m_0 and m_1, in a LR query to obtain the challenge encryption $c^* = \mathtt{Enc}(pk^*, m_b; f(\mathtt{Enc}(pk, m; r))$, where pk^* is the challenge public key, b is the challenge bit, and r is the random value chosen in the IND-RRR-CCA game. Then, by submitting $(pk, m, \mathtt{id})$ to his encryption oracle ENC, the adversary will obtain $c_r = \mathtt{Enc}(pk, m; r)$ and can compute $\tilde{r} = f(c_r)$. Finally, the adversary can compute $c_0 = \mathtt{Enc}(pk^*, m_0; \tilde{r})$ and $c_1 = \mathtt{Enc}(pk^*, m_1; \tilde{r})$, and by testing whether $c_0 = c^*$ or $c_1 = c^*$, he will learn the challenge bit b. □ (**Theorem 4**)

Note that the only functions required in the above attack, are $f(\mathtt{Enc}(pk, m, \cdot))$ and $\mathtt{id}(\cdot)$. These functions are easily seen to be output-unpredictable assuming the underlying encryption scheme in the construction is IND-CPA secure, and that an appropriate mapping function f is chosen. They can likewise be seen to be collision-resistant under the same assumptions. Furthermore, it should be noted that the above theorem does not require the REFRESH oracle to be queried, and hence is also true for the IND-RR-CCA notion defined in [34].

While the above theorem holds for all encryption schemes in general, stronger results might hold for concrete schemes. In particular, even if $f(\mathtt{Enc}(pk, m; \cdot)) \notin \Phi$, the structure of a concrete scheme might still allow an adversary to mount a similar attack to the above based on multiple queries to his LR and ENC oracles, for carefully selected functions. However, the IND-RRR-CCA security notion bounds the information an adversary can extract before the randomness is refreshed, which will allow us to construct a generic conversion of a PKE scheme achieving IND-RRR-CCA security for relatively large and complex function classes Φ and Ψ. Interestingly, the above theorem furthermore illustrates some of the limitations of the building blocks used in [34] to achieve related randomness security; see the full version of the paper for a brief discussion of this.

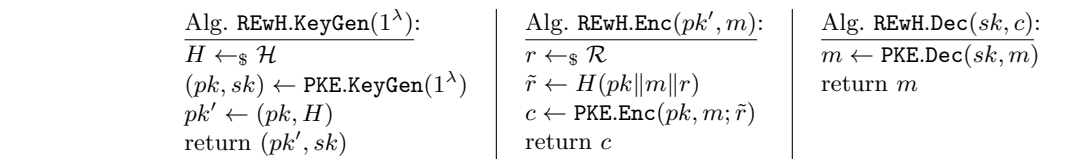

Fig. 4. Scheme REwH constructed from a PKE scheme PKE and a hash family $\mathcal{H}$.

5 On the IND-RR-CCA Security of REwH in the Random Oracle Model

In this section, we will revisit the IND-RR-CCA security of the REwH (Randomized-Encrypt-with-Hash) scheme in the random oracle model.

The REwH scheme was introduced by Bellare *et al.* [6] to hedge against randomness failures, and was furthermore studied by Ristenpart and Yilek [39] in the context of virtual machine reset attacks. The basic idea of the scheme is to modify the encryption function of an existing encryption scheme to use randomness derived by hashing all the inputs to the encryption algorithm: the public key, the message, and the randomness. Assuming the hash function is a random oracle, the scheme will remain secure (in the sense of the security of the underlying encryption scheme), as long as this triple of inputs remains unpredictable to the adversary. The scheme is shown in Figure 4.

In [34], Paterson *et al.* showed that this scheme is additionally Φ-IND-RR-ATK secure assuming the underlying encryption scheme is IND-ATK secure, where ATK is either CPA or CCA, and Φ is both output-unpredictable and collision-resistant. Considering the impossibility result in the previous section, this might initially appear somewhat surprising. However, as already mentioned, the results in [34] implicitly assume that the functions in Φ are *independent* of the used random oracle i.e. the functions in Φ cannot capture the encryption function $\mathtt{Enc}(pk, m; r) = \mathtt{Enc}'(pk, m; H(pk, m, r))$ of the REwH construction, where $\mathtt{Enc}'$ is the encryption function of the underlying encryption scheme.

In this section, we will consider Φ which might depend on the random oracle, i.e. we will assume that functions in Φ might access the random oracle. This is reminiscent of Albrecht *et al.* [5], who considered RKA-security of symmetric encryption in the ideal cipher model with RKA-functions that depend on the ideal cipher. To show security in this stronger setting, we need to place additional restrictions on the adversary (as shown by the direct attack in the previous section). Here, we will consider the following limitation of the adversary's queries.

Definition 12 (Indirect H-query uniqueness) *Consider an adversary $\mathcal{A}$ interacting in the Φ-IND-RR-CCA security game in the random oracle model. $\mathcal{A}$ is said to respect* indirect H-query uniqueness *if, all random oracle queries caused by $\mathcal{A}$'s queries to his* Enc *and* LR *oracles, are unique.*

Note that, in the above definition, $\mathcal{A}$ is not restricted in terms of his queries directly to the random oracle; only the indirect queries caused by $\mathcal{A}$'s Enc and

LR queries are restricted. With this definition in place, we can now show the following result for the REwH construction.

Theorem 5 *Let* PKE *be an* IND-CCA *secure PKE scheme, and let* $\Phi = \{\phi : \mathcal{R} \to \mathcal{R}\}$*, be an output-unpredictable function family, where* $\mathcal{R} = \mathcal{R}_\lambda$ *is the randomness space of* PKE.Enc*. Then the* REwH *scheme based on* PKE *is* Φ-IND-RR-CCA *secure against adversaries respecting indirect H-query uniqueness, assuming the hash function in the* REwH *construction is modeled as a random oracle. More precisely, for all equality and indirect H-query uniqueness respecting adversaries* $\mathcal{A}$ *making* $q_{lr} = q_{lr}(\lambda)$ LR *queries,* $q_{enc} = q_{enc}(\lambda)$ ENC *queries, and* $q_{RO} = q_{RO}(\lambda)$ *random oracle queries, there exists an algorithm* $\mathcal{B}$ *such that*

$$\mathtt{Adv}^{\mathtt{IND\text{-}RR\text{-}CCA}}_{\mathtt{REwH},\mathcal{A}}(\lambda) \leq \mathtt{Adv}^{\mathtt{IND\text{-}CCA}}_{\mathtt{PKE},\mathcal{B}}(\lambda) + 2q_{RO}(q_{lr} + q_{enc}) \cdot \mathsf{UP}^{\Phi}(\lambda).$$

The proof of the above theorem can be found in the full version of the paper.

Note that in the above theorem, collision resistance of Φ is not required. This is because the indirect H-query uniqueness property will prevent an adversary from submitting functions ϕ_1, ϕ_2 to his ENC and LR oracles, for which a collision $\phi_1(r) = \phi_2(r)$ occurs, assuming the queried public keys and messages are the same. (If the submitted public keys and messages are different, indirect H-query uniqueness will not imply that a collision cannot occur, but this will not affect the proof, since the inputs to the random oracle will remain distinct.)

The requirement that the adversary is indirect H-query uniqueness respecting might be considered to be somewhat artificial, in that there seems to be no reasonable argument for this assumption to hold for adversaries in the practical settings in which related randomness attacks might be a concern. In the following sections, we will explore the possibilities of achieving security in the standard model, under the arguably realistic assumption that the adversary can only mount a limited number of queries before new entropy is added to the system on which encryption is being done.

6 Bounded RKA and Correlated-Input Security from t-wise Independent Hash Functions

In this section, we show how to construct the building blocks needed for our standard-model IND-RRR-CCA-secure PKE scheme. More concretely, we will start out by showing a key-dependent variant of the leftover hash lemma for correlated inputs. This, in turn, allows us to show that a family of t-wise independent hash functions leads to a bounded correlated-input secure function family, in the sense that a bound for the number q of correlated inputs must be known a priori. Finally, we will then show how a PRF (with public parameters) that provides RKA-security as long as an adversary makes at most q related key derivation queries, can be constructed from an ordinary PRF and a q bounded correlated-input secure function family. This type of PRF will be used to construct our IND-RRR-CCA-secure PKE scheme in Section 7. We believe that each of the intermediate results might find other applications than the construction of related randomness secure PKE scheme, and hence, might be of independent interest.

6.1 Key-Dependent Leftover Hash Lemma for Correlated Inputs

The ordinary leftover hash lemma [28] requires that the input to the hash function is chosen independently of the description of the hash function (i.e. the hash key). The first key-dependent versions of the leftover hash lemma were shown in [21, 41], and was extended to consider leakage in [14]. A "crooked" version for block sources was shown in [38].

The version of the leftover hash lemma that we will show in the following, differs from the previous work in that we consider unrestricted inputs which can both be arbitrarily correlated and key-dependent. Our lemma is as follows.

Lemma 1 *Let $\mathcal{H} : D \to R$ be a family of t-wise independent hash functions where $t > 8$ is an even number, and let $\mathcal{X}$ be a family of collections of q (correlated) random variables $\boldsymbol{X} = (X_1, \ldots, X_q)$ over D, such that $\mathtt{H}_\infty(X_i) \geq \gamma$ for all $1 \leq i \leq q$, and $\Pr[X_i = X_j] = 0$ for all $1 \leq i \neq j \leq q$. Furthermore, let $\epsilon, \delta > 0$ be such that*

$$t \geq \log|\mathcal{X}| + q\log|R| + \log\frac{1}{\delta}, \quad \textit{and} \quad \gamma \geq q\log|R| + 2\log\frac{1}{\epsilon} + \log t + 2. \tag{1}$$

Then, with probability $1 - \delta$ over the choice of $H \leftarrow_\$ \mathcal{H}$,

$$\Delta[H(\boldsymbol{X}), (\underbrace{U_R, \ldots, U_R}_{q})] \leq \epsilon$$

holds for all $\boldsymbol{X} \in \mathcal{X}$, where U_R denotes the uniform distribution on R.

Proof. (of Lemma 1) We start by considering a fixed collection of random variables $\boldsymbol{X} = (X_1, \ldots, X_q)$ such that $\mathtt{H}_\infty(X_i) \geq \gamma$ for all $1 \leq i \leq q$ and $\Pr[X_i = X_j] = 0$ for all $1 \leq i \neq j \leq q$, and a fixed value $\boldsymbol{y} \in R^q$. Note that the condition of $\boldsymbol{X}$ implies that every coordinate of (an outcome of) $\boldsymbol{X}$ is always distinct. Therefore, due to the t-wise independence of $\mathcal{H}$, and that $q < t$, we must have that, for any $\boldsymbol{x}$ in the support of $\boldsymbol{X}$ (which is a subset of D^q),

$$\Pr_{H \leftarrow_\$ \mathcal{H}}[H(\boldsymbol{x}) = \boldsymbol{y}] = \frac{1}{|R|^q}. \tag{2}$$

Now let $I_{H(\boldsymbol{x})=\boldsymbol{y}}$ be the indicator variable that takes on the value 1 if $H(\boldsymbol{x}) = \boldsymbol{y}$ (and 0 otherwise), and let $p_{\boldsymbol{x}} = \Pr[\boldsymbol{X} = \boldsymbol{x}] \cdot I_{H(\boldsymbol{x})=\boldsymbol{y}}$ and $p = \sum_{\boldsymbol{x} \in D^q} p_{\boldsymbol{x}}$. The expected value of p (over the choice $H \leftarrow_\$ \mathcal{H}$) is then

$$\mathtt{E}[p] = \mathtt{E}[\sum_{\boldsymbol{x} \in D^q} p_{\boldsymbol{x}}] = \sum_{\boldsymbol{x} \in D^q} \Pr[\boldsymbol{X} = \boldsymbol{x}] \cdot \mathtt{E}[I_{H(\boldsymbol{x})=\boldsymbol{y}}] = \frac{1}{|R|^q},$$

where the last equality follows from $\mathtt{E}[I_{H(\boldsymbol{x})=\boldsymbol{y}}] = \Pr_{H \leftarrow_\$ \mathcal{H}}[H(\boldsymbol{x}) = \boldsymbol{y}] = |R|^{-q}$, which in turn follows from Equation (2). Finally let $P_{\boldsymbol{x}} = 2^\gamma \cdot p_{\boldsymbol{x}}$ and

$$P = \sum_{\boldsymbol{x} \in D^q} P_{\boldsymbol{x}} = 2^\gamma p.$$

The expected value of P must then be $\mathtt{E}[P] = 2^\gamma \cdot \mathtt{E}[p] = 2^\gamma \cdot |R|^{-q}$.

We will now apply the tail bound from Theorem 1 to P and $\mathtt{E}[P]$ (note that the $P_{\boldsymbol{x}}$ values are t-wise independent due to $\mathcal{H}$ (and thereby also $I_{H(\boldsymbol{x})=\boldsymbol{y}}$) being t-wise independent over the choice of H). Doing so yields

$$\begin{aligned}\Pr_{H \leftarrow_\$ \mathcal{H}}[|P - \mathtt{E}[P]| \geq \epsilon \cdot \mathtt{E}[P]] &\leq \left(\frac{t \cdot |R|^q}{\epsilon^2 \cdot 2^\gamma}\right)^{\frac{t}{2}} \\ &= \left(\frac{1}{2^{\gamma - 2\log\frac{1}{\epsilon} - \log t - q\log|R|}}\right)^{\frac{t}{2}} \\ &\leq 2^{-t},\end{aligned}$$

where the last inequality follows from the bound on $\log|R|$ given in the theorem. Note that, due to the definition of P and p, we now have that, for any $\epsilon > 0$,

$$\begin{aligned}\Pr_{H \leftarrow_\$ \mathcal{H}}\left[\left|\Pr_{\boldsymbol{x} \leftarrow \boldsymbol{X}}[H(\boldsymbol{x}) = \boldsymbol{y}] - \frac{1}{|R|^q}\right| \geq \frac{\epsilon}{|R|^q}\right] &= \Pr_{H \leftarrow_\$ \mathcal{H}}\left[\left|p - \frac{1}{|R|^q}\right| \geq \epsilon \cdot \frac{1}{|R|^q}\right] \\ &= \Pr_{H \leftarrow_\$ \mathcal{H}}[|P - \mathtt{E}[P]| \geq \epsilon \cdot \mathtt{E}[P]] \\ &\leq 2^{-t}.\end{aligned}$$

The above inequality holds for any value $\boldsymbol{y} \in R^q$ and any set $\boldsymbol{X} = (X_1, \ldots, X_q)$ of random variables over D^q, satisfying the criteria given in the theorem. Taking the union bound over all possible $\boldsymbol{y} \in R^q$ values and all collections $\boldsymbol{X} \in \mathcal{X}$, yields that with probability $1 - |\mathcal{X}| \cdot |R|^q \cdot 2^{-t}$ over the choice of H, we have that $|\Pr[H(\boldsymbol{x}) = \boldsymbol{y}] - |R|^{-q}| \leq \epsilon|R|^{-q}$ for *all* choices of $\boldsymbol{y} \in R^q$ and $\boldsymbol{X} \in \mathcal{X}$. This immediately implies that the statistical distance between $H(\boldsymbol{X})$ and the uniform distribution over R^q, is at most ϵ.

Finally, setting $t \geq \log|\mathcal{X}| + q\log|R| + \log 1/\delta$ ensures that $\delta \geq |\mathcal{X}| \cdot |R|^q \cdot 2^{-t}$, as required. □ (**Lemma 1**)

6.2 Correlated-Input Secure Functions

Firstly, we will formalize the security notion *correlated-input pseudorandomness* (CIPR).

Definition 13 *Let $\mathcal{H} = \{H : D \to R\}$ be a family of (hash) functions with domain $D = D_\lambda$ and range $R = R_\lambda$, $\Phi = \{\phi : D \to D\}$ be a function family, and $q = q(\lambda)$ be a positive polynomial. Then, for an adversary $\mathcal{A}$, consider the security game shown in Figure 5. In the game, it is required that all queries ϕ submitted by $\mathcal{A}$ belong to Φ, and must be distinct with each other. The advantage of the adversary $\mathcal{A}$ interacting with the security game with respect to $\mathcal{H}$, is defined to be*

$$\mathtt{Adv}^{\mathtt{CIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda) = 2\left|\Pr[\mathrm{CIPR}^{\mathcal{A},\Phi}_{\mathcal{H},q}(\lambda) \Rightarrow 1] - \frac{1}{2}\right|.$$

$\mathcal{H}$ is said to be (q,Φ)-CIPR secure, if for all PPT adversaries $\mathcal{A}$, $\mathtt{Adv}^{\mathtt{CIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda)$ is negligible in the security parameter λ.

$\underline{\mathrm{CIPR}^{\mathcal{A},\Phi}_{\mathcal{H},q}(\lambda)}$:
$H \leftarrow_{\$} \mathcal{H}$
$x \leftarrow_{\$} D$
$b \leftarrow_{\$} \{0,1\}$
$queries \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\mathrm{HASH}}(1^\lambda, H)$
return $(b = b')$

$\underline{\text{proc. } \mathrm{HASH}(\phi)}$:
if $queries > q$
  return $\perp$
if $b = 0$
  $h \leftarrow H(\phi(x))$
else
  $h \leftarrow_{\$} R$
$queries \leftarrow queries + 1$
return h

Fig. 5. Game defining correlated-input pseudorandomness (CIPR) of a hash family $\mathcal{H}$.

The following theorem shows that a t-wise independent hash function family satisfies the above defined CIPR notion.

Theorem 6 (Correlated-Input Pseudorandomness of t-wise Independent Hash Functions) *Let $t = t(\lambda)$, $p = p(\lambda)$, and $q = q(\lambda)$ be integer-valued positive polynomials such that t is always even and larger than 8. Let $\mathcal{H} = \{H : D \to R\}$ be a family of t-wise independent hash functions with domain $D = D_\lambda$ and range $R = R_\lambda$, let $\Phi = \{\phi : D \to D\}$ be a function family such that $|\Phi| \le 2^p$, and let $\mathsf{CR}^{\Phi}(\lambda) \le 1/(2\binom{q}{2})$. Furthermore, let $\epsilon = \epsilon(\lambda)$ and $\delta = \delta(\lambda)$ be any functions such that their range is $[0,1]$ and satisfy:*

$$t \ge q\cdot(p+\log|R|)+\log\frac{1}{\delta} \quad \textit{and} \quad \log\frac{1}{\mathsf{UP}^{\Phi}(\lambda)} \ge q\log|R|+2\log\frac{1}{\epsilon}+\log t+3. \tag{3}$$

Then, for all computationally unbounded adversaries $\mathcal{A}$ that make at most q queries, we have

$$\mathtt{Adv}^{\mathtt{CIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda) \le 2\cdot|R|^{q-1}\cdot(\epsilon+\delta+\tbinom{q}{2}\cdot\mathsf{CR}^{\Phi}(\lambda)).$$

The above theorem immediately gives us the following corollary:

Corollary 1 *Let $t = t(\lambda)$, $p = p(\lambda)$, and $q = q(\lambda)$ be integer-valued positive polynomials such that t is always even and larger than 8. Let $\mathcal{H} = \{H : D \to R\}$ be a family of t-wise independent hash functions with domain $D = D_\lambda$ and range $R = R_\lambda$ such that $|D| \ge |R| = O(2^\lambda)$. Let $\Phi = \{\phi : D \to D\}$ be a function family such that $|\Phi| \le 2^p$. Assume that*

$$\begin{aligned} t &\ge pq + (2q-1)\log|R| + \lambda, \\ \mathsf{UP}^{\Phi}(\lambda) &\le |R|^{-(3q-2)}\cdot 2^{-(2\lambda+O(\log\lambda))}, \\ \mathsf{CR}^{\Phi}(\lambda) &\le \tbinom{q}{2}^{-1}\cdot|R|^{-(q-1)}\cdot 2^{-\lambda}. \end{aligned} \tag{4}$$

Then, for all computationally unbounded adversaries $\mathcal{A}$ that make at most q queries, and for sufficiently large λ, we have

$$\mathtt{Adv}^{\mathtt{CIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda) \le 6\cdot 2^{-\lambda}.$$

Proof. (of Corollary 1) We set $\epsilon = \delta = |R|^{-(q-1)} \cdot 2^{-\lambda}$ in Theorem 6. Then, the assumption on t in Equation (4) implies the condition required for t in Equation (3). Furthermore, since p, q, and $\log |R|$ are all polynomials of λ, we have $\log t = O(\log \lambda)$. This fact, combined with the assumption on $\mathsf{UP}^{\Phi}(\lambda)$ in Equation (4), implies that $\mathsf{UP}^{\Phi}(\lambda)$ satisfies the condition required for it in Equation (3) for all sufficiently large λ. Therefore, we can now invoke Theorem 6: for all computationally unbounded adversaries $\mathcal{A}$ that make at most q queries, and for all sufficiently large λ, we have

$$\begin{aligned}\mathtt{Adv}^{\mathtt{CIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda) &\leq 2 \cdot |R|^{q-1} \cdot (\epsilon + \delta + \tbinom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda)) \\ &\leq 2 \cdot |R|^{q-1} \cdot (|R|^{-(q-1)} \cdot 2^{-\lambda} + |R|^{-(q-1)} \cdot 2^{-\lambda} + |R|^{-(q-1)} \cdot 2^{-\lambda}) \\ &= 6 \cdot 2^{-\lambda},\end{aligned}$$

as required. □ (**Corollary 1**)

Now, we proceed to the proof of Theorem 6. The proof consists of two steps. Firstly, we will make use of our variant of the leftover hash lemma (Lemma 1) to show that a t-wise independent hash functions $\mathcal{H}$ satisfies a weaker "non-adaptive" version of correlated-input pseudorandomness, which we denote `naCIPR`, in which an adversary has to submit all of his hash queries at once parallelly. Then we make use of complexity leveraging to move from `naCIPR` security to the full `CIPR` security (this step causes the loss factor $|R|^{q-1}$ appearing in the upperbound of an adversary's advantage shown in the theorem).

Proof. (of Theorem 6) We firstly consider the "non-adaptive" version of the `CIPR` game shown in Fig. 5, in which an adversary $\mathcal{A}$ has to submit its hash queries non-adaptively (i.e. parallelly). That is, an adversary $\mathcal{A}$, on input 1^{λ} and H, submits a set of functions $(\phi_i)_{i \in [q]}$ all at once to the hash oracle Hash, and receives the set of answers $(h_i)_{i \in [q]}$ where each h_i is either the real hash value $H(\phi_i(x))$ or a random value chosen uniformly from the range R of H. Let us denote by $\mathtt{Adv}^{\mathtt{naCIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}$ the advantage of an adversary $\mathcal{A}$ in this game.

By using Lemma 1, we show that the advantage of any computationally unbounded non-adaptive adversary, is bounded as stated in the following lemma:

Lemma 2 *Under the same setting as in Theorem 6, for all computationally unbounded adversaries $\mathcal{A}$ that make at most $q = q(\lambda)$ queries, we have*

$$\mathtt{Adv}^{\mathtt{naCIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda) \leq 2 \left(\epsilon + \delta + \tbinom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda) \right). \tag{5}$$

Proof. (of Lemma 2) We first introduce several necessary definitions: for a security parameter λ, a hash function $H \in \mathcal{H}$, and a deterministic non-adaptive adversary $\mathcal{A}$ that runs in the `naCIPR` game and makes q queries, let $(\phi_1, \ldots, \phi_q)$ be the functions submitted by $\mathcal{A}(1^{\lambda}, H)$ in $\mathcal{A}$'s non-adaptive parallel query. [7]

[7] We will later show an upperbound of the advantage for all *computationally unbounded* non-adaptive adversaries $\mathcal{A}$ in the `naCIPR` game, in which case considering whether $\mathcal{A}$ is deterministic or probabilistic does not matter because a computationally unbounded adversary can find its best randomness and use this. Hence, considering only deterministic adversaries here is sufficient for our purpose.

Note that since we are considering a deterministic adversary $\mathcal{A}$, once we fix $\mathcal{A}$ and $H \in \mathcal{H}$, the functions $(\phi_1, \ldots, \phi_q)$ are determined without any ambiguity.

Let $\mathsf{NoColl}_{\mathcal{A},H} \subseteq D$ be the subset of D that consists of "collision-free" elements with respect to $\mathcal{A}$ and H, in the following sense:

$$\mathsf{NoColl}_{\mathcal{A},H} := \Big\{\ x \in D \ \Big|\ \forall i, j \in [q] \text{ s.t. } i \neq j : \phi_i(x) \neq \phi_j(x)\ \Big\},$$

where each ϕ_i is the i-th function that appears in $\mathcal{A}$'s parallel query on input $(1^\lambda, H)$. Note that if we pick $x \in D$ uniformly at random, the probability that $\phi_i(x) = \phi_j(x)$ occurs for some (i, j) with $1 \leq i \neq j \leq q$ is upperbounded by $\binom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda)$. This implies $\Pr_{x \leftarrow_\$ D}[x \in \mathsf{NoColl}_{\mathcal{A},H}] \geq 1 - \binom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda)$. Equivalently, we have

$$|\mathsf{NoColl}_{\mathcal{A},H}| \geq (1 - \tbinom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda)) \cdot |D| \geq \frac{1}{2} \cdot |D|, \tag{6}$$

where in the last inequality we use $\mathsf{CR}^{\Phi}(\lambda) \leq 1/(2\binom{q}{2})$.

Then, we define the random variable $\boldsymbol{X}_{\mathcal{A},H} = (X_1, \ldots, X_q)$, defined over D^q, as follows:

$$\boldsymbol{X}_{\mathcal{A},H} = (X_1, \ldots, X_q) := \Big\{ x \leftarrow_\$ \mathsf{NoColl}_{\mathcal{A},H};\ \forall i \in [q] : x_i \leftarrow \phi_i(x) : (x_1, \ldots, x_q) \Big\}. \tag{7}$$

We then define $\mathcal{X}$ to be the set consisting of the random variables $\boldsymbol{X}_{\mathcal{A},H}$ for all possible deterministic non-adaptive adversaries $\mathcal{A}$ and all hash functions $H \in \mathcal{H}$. Namely, we define

$$\mathcal{X} := \bigcup_{\mathcal{A}} \Big\{\ \boldsymbol{X}_{\mathcal{A},H}\ \Big|H \in \mathcal{H}\Big\}, \tag{8}$$

where the union is taken over all possible non-adaptive adversaries $\mathcal{A}$.

We note that each ϕ_i in an adversary $\mathcal{A}$'s parallel query belongs to the set Φ (no matter what the adversary $\mathcal{A}$ is and no matter what hash function $H \in \mathcal{H}$ $\mathcal{A}$ receives), and note also that $|\Phi| \leq 2^q$ holds. Therefore, the number of distinct random variables $\boldsymbol{X}_{\mathcal{A},H}$ is at most 2^{pq}, namely, we have $|\mathcal{X}| \leq 2^{pq}$. Furthermore, note also that by definition, we have $\Pr[X_i = X_j] = 0$ for all $i \neq j \in [q]$ and all $\boldsymbol{X}_{\mathcal{A},H} = (X_1, \ldots, X_q) \in \mathcal{X}$ (no matter what $\mathcal{A}$ is and no matter what hash function $H \in \mathcal{H}$ $\mathcal{A}$ receives).

We now consider the min-entropy of each coordinate X_i of the random variables $\boldsymbol{X}_{\mathcal{A},H} \in \mathcal{X}$. By applying the lemma by Dodis and Yu [22, Lemma 1] and Equation (6), for every $\phi \in \Phi$ and $y \in D$, we have

$$\Pr_{x \leftarrow_\$ \mathsf{NoColl}_{\mathcal{A},H}}[\phi(x) = y] \leq \frac{|D|}{|\mathsf{NoColl}_{\mathcal{A},H}|} \cdot \Pr_{x \leftarrow_\$ D}[\phi(x) = y] \leq 2 \cdot \Pr_{x \leftarrow_\$ D}[\phi(x) = y]. \tag{9}$$

Furthermore, by definition $\max_{y \in D}\{\Pr_{x \leftarrow_\$ D}[\phi(x) = y]\} \leq \mathsf{UP}^{\Phi}(\lambda)$ holds for every $\phi \in \Phi$. By combining this with Equation (9), for every $i \in [q]$, we have

$$\begin{aligned} \mathtt{H}_\infty(X_i) &= -\log\Big(\max_{y \in D}\Big\{\Pr_{x \leftarrow_\$ \mathsf{NoColl}_{\mathcal{A},H}}[\phi_i(x) = y]\Big\}\Big) \\ &\geq -\log\left(\max_{y \in D}\Big\{2 \cdot \Pr_{x \leftarrow_\$ D}[\phi_i(x) = y]\Big\}\right) \geq \log \frac{1}{2\mathsf{UP}^{\Phi}(\lambda)}. \end{aligned} \tag{10}$$

In words, we have seen that for all random variables $\boldsymbol{X} = (X_1, \ldots, X_q) \in \mathcal{X}$, the min-entropy of each X_i is lowerbounded by $\log(1/2\mathsf{UP}^{\Phi}(\lambda))$.

For a number $\epsilon' > 0$, define the set $\mathsf{GoodHash}_{\epsilon'} \subseteq \mathcal{H}$ by

$$\mathsf{GoodHash}_{\epsilon'} := \left\{ H \in \mathcal{H} \;\middle|\; \forall \boldsymbol{X} \in \mathcal{X} : \Delta[H(\boldsymbol{X}), (\underbrace{U_R, \ldots, U_R}_{q})] \leq \epsilon' \right\}.$$

Recall that $|\mathcal{X}| \leq 2^{pq}$. Hence, by Equation (10), if $\delta' > 0$ is a number such that

$$t \geq q \cdot (\log |R| + p) + \log \frac{1}{\delta'} \quad \text{and} \quad \log \frac{1}{2 \cdot \mathsf{UP}^{\Phi}(\lambda)} \geq q \log |R| + 2 \log \frac{1}{\epsilon'} + \log t + 2,$$

then the condition on t in Equation (1) in Lemma 1 is satisfied. Furthermore, due to Equation (10) and the assumption on $\log(1/\mathsf{UP}^{\Phi}(\lambda))$ in Lemma 2, all random variables $\boldsymbol{X} = (X_1, \ldots, X_q) \in \mathcal{X}$ satisfy the second condition (i.e. the lowerbound on the min-entropy in each entry X_i) in Equation (1). Hence, by applying Lemma 1 to the set of variables $\mathcal{X}$ (which we have seen satisfies all the requirements for Lemma 1), we have $|\mathsf{GoodHash}_{\epsilon'}| \geq (1 - \delta') \cdot |\mathcal{H}|$.

Having defined the things we need, we are now ready to show an upperbound on the advantage of all non-adaptive adversaries $\mathcal{A}$ in the `naCIPR` game. Fix arbitrarily a computationally unbounded adversary $\mathcal{A}$ that makes at most q queries in the `naCIPR` game. Fix also arbitrarily functions $\epsilon = \epsilon(\lambda)$ and $\delta = \delta(\lambda)$ satisfying Equation (3). Our goal is to show that Equation (5) is satisfied for the above $\mathcal{A}$, and numbers $\epsilon' = \epsilon$, and $\delta' = \delta$.

Let S be the event that $\mathcal{A}$ succeeds in guessing its challenge bit (i.e. $b' = b$ occurs), and let GH (which stands for "**G**ood **H**ash") be the event that the hash function H that $\mathcal{A}$ receives satisfies $H \in \mathsf{GoodHash}_{\epsilon}$, and let NC (which stands for "**N**o **C**ollision") be the event that there exist no indices $i, j \in [q]$ such that $\phi_i(x) = \phi_j(x)$, where $x \in D$ is the value chosen randomly at the non-adaptive game, and ϕ_i (resp. ϕ_j) be the i-th (resp. j-th) function in the parallel query $(\phi_1, \ldots, \phi_q)$ submitted by $\mathcal{A}$ on input $(1^{\lambda}, H)$.

We proceed to estimating lower and upperbounds for $\Pr[\mathsf{S}]$. On the one hand, we have

$$\begin{aligned}
\Pr[\mathsf{S}] &\geq \Pr[\mathsf{S} \wedge \mathsf{GH} \wedge \mathsf{NC}] \\
&= \Pr[\mathsf{S} | \mathsf{GH} \wedge \mathsf{NC}] \cdot \Pr[\mathsf{GH} \wedge \mathsf{NC}] \\
&= \Pr[\mathsf{S} | \mathsf{GH} \wedge \mathsf{NC}] \cdot (1 - \Pr[\overline{\mathsf{GH}} \vee \overline{\mathsf{NC}}]) \\
&\geq \Pr[\mathsf{S} | \mathsf{GH} \wedge \mathsf{NC}] - \Pr[\overline{\mathsf{GH}}] - \Pr[\overline{\mathsf{NC}}]. \qquad (11)
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
\Pr[\mathsf{S}] &= \Pr[\mathsf{S} \wedge \mathsf{GH} \wedge \mathsf{NC}] + \Pr[\mathsf{S} \wedge (\overline{\mathsf{GH}} \vee \overline{\mathsf{NC}})] \\
&\leq \Pr[\mathsf{S} | \mathsf{GH} \wedge \mathsf{NC}] + \Pr[\overline{\mathsf{GH}} \vee \overline{\mathsf{NC}}] \\
&\leq \Pr[\mathsf{S} | \mathsf{GH} \wedge \mathsf{NC}] + \Pr[\overline{\mathsf{GH}}] + \Pr[\overline{\mathsf{NC}}]. \qquad (12)
\end{aligned}$$

Here, by definition, we have $\Pr[\mathsf{GH}] \geq 1 - \delta$ and $\Pr[\mathsf{NC}] \geq 1 - \binom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda)$, where the probabilities in the left hand side of both of the inequalities are over the $\mathtt{naCIPR}$ game. Furthermore, the event S conditioned on GH and NC, corresponds to the situation where $\mathcal{A}$, on input 1^λ and $H \in \mathsf{GoodHash}_\epsilon$, receives $(h_1, \ldots, h_q)$ that is sampled from either the distribution $H(\boldsymbol{X}_{\mathcal{A},H})$ where $\boldsymbol{X}_{\mathcal{A},H} \in \mathcal{X}$ or the uniform distribution $(U_R)^q$ over R^q, and succeeds in guessing which is the case. Here, due to the definitions of $\mathsf{GoodHash}_\epsilon$ and $\boldsymbol{X}_{\mathcal{A},H}$, the statistical distance between $H(\boldsymbol{X}_{\mathcal{A},H})$ and the uniform distribution $(U_R)^q$ is at most ϵ. Hence, we have

$$\frac{1}{2} - \epsilon \leq \Pr[\mathsf{S} | \mathsf{GH} \wedge \mathsf{NC}] \leq \frac{1}{2} + \epsilon.$$

Combining these inequalities with Equations (11) and (12), we obtain

$$-(\epsilon + \delta + \tbinom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda)) \leq \Pr[\mathsf{S}] - \frac{1}{2} \leq \epsilon + \delta + \tbinom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda),$$

which implies

$$\mathtt{Adv}^{\mathtt{naCIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda) = 2\left|\Pr[\mathsf{S}] - \frac{1}{2}\right| \leq 2\left(\epsilon + \delta + \tbinom{q}{2} \cdot \mathsf{CR}^{\Phi}(\lambda)\right),$$

as required. □ (**Lemma 2**)

Finally, as the last step of the proof of Theorem 6, we show that by a complexity leveraging argument, ordinary (adaptive) correlated-input pseudorandomness is implied by its non-adaptive version. More precisely, we show the following lemma:

Lemma 3 *Let $q = q(\lambda)$ be a positive polynomial. Let $\mathcal{H} = \{H : D \to R\}$ and $\Phi = \{\phi : D \to D\}$ be families of functions with domain $D = D_\lambda$ and ranges $R = R_\lambda$ and D, respectively. Then, for all computationally unbounded adversaries $\mathcal{A}$ that make q queries, there exists a computationally unbounded non-adaptive adversary $\mathcal{B}$ that makes q queries, such that*

$$\mathtt{Adv}^{\mathtt{naCIPR}}_{\mathcal{H},q,\mathcal{B},\Phi}(\lambda) = \frac{1}{|R|^{q-1}} \cdot \mathtt{Adv}^{\mathtt{CIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda). \tag{13}$$

Proof. (of Lemma 3) Fix arbitrarily a positive polynomial q and a computationally unbounded adversary $\mathcal{A}$ that runs in the $\mathtt{CIPR}$ game and makes q queries. Using $\mathcal{A}$ as a building block, we show how to construct another computationally unbounded adversary $\mathcal{B}$ that runs in the $\mathtt{naCIPR}$ game, makes in exactly the same number of queries as $\mathcal{A}$, and has the advantage as stated in Equation (13). The description of $\mathcal{B}$ is as follows:

$\mathcal{B}(1^\lambda, H)$**:** $\mathcal{B}$ first chooses $q-1$ values $h'_1, \ldots, h'_{q-1} \leftarrow_{\$} R$ uniformly at random, and runs $\mathcal{A}(1^\lambda, H)$, where $\mathcal{B}$ answers to $\mathcal{A}$'s i-th query ϕ_i by h'_i (no matter what ϕ_i is). When $\mathcal{A}$ makes the q-th query ϕ_q, $\mathcal{B}$ submits q functions $(\phi_i)_{i\in[q]}$ as its "parallel" query to $\mathcal{B}$'s hash oracle, and receives the results $(h^*_i)_{i\in[q]}$. Then, $\mathcal{B}$ proceeds as follows:

– If $h_i^* = h'_i$ holds for all $i \in [q-1]$, then $\mathcal{B}$ finds that its simulation for $\mathcal{A}$ was "good", and returns h_q^* as the answer to $\mathcal{A}$'s q-th query. When $\mathcal{A}$ terminates with output b', $\mathcal{B}$ sets $\sigma' \leftarrow b'$.
– Otherwise (i.e. $h_i^* \neq h'_i$ holds for some $i \in [q-1]$), $\mathcal{B}$ decides that it does not use $\mathcal{A}$'s output, and sets $\sigma' \leftarrow_{\$} \{0,1\}$ uniformly at random.

Finally, $\mathcal{B}$ terminates with output σ'.

The above completes the description of $\mathcal{B}$. Let σ be $\mathcal{B}$'s challenge bit in its non-adaptive game. Furthermore, let S be the event that $\sigma' = \sigma$ occurs, and G be the event that $h_i^* = h'_i$ holds for all $i \in [q-1]$ (where both of the events are defined in $\mathcal{B}$'s `naCIPR` game). By definition, $\mathcal{B}$'s advantage in the `naCIPR` game can be estimated as follows:

$$\begin{aligned}
\mathtt{Adv}^{\mathtt{naCIPR}}_{\mathcal{H},q,\mathcal{B},\Phi}(\lambda) &= 2\left|\Pr[\mathsf{S}] - \frac{1}{2}\right| \\
&= 2\left|\Pr[\mathsf{S}|\mathsf{G}]\cdot\Pr[\mathsf{G}] + \Pr[\mathsf{S}|\overline{\mathsf{G}}]\cdot\Pr[\overline{\mathsf{G}}] - \frac{1}{2}(\Pr[\mathsf{G}] + \Pr[\overline{\mathsf{G}}])\right| \\
&= 2\left|\Pr[\mathsf{G}]\cdot(\Pr[\mathsf{S}|\mathsf{G}] - \frac{1}{2}) + \Pr[\overline{\mathsf{G}}]\cdot(\Pr[\mathsf{S}|\overline{\mathsf{G}}] - \frac{1}{2})\right|. \qquad (14)
\end{aligned}$$

Now, since all $\{h'_i\}_{i\in[q-1]}$ are chosen uniformly at random, independently of $\mathcal{A}$'s behavior and $\mathcal{B}$'s challenge bit, we have $\Pr[\mathsf{G}] = 1/|R|^{q-1}$. Moreover, once G occurs, $\mathcal{B}$ simulates the `CIPR` game perfectly for $\mathcal{A}$ so that $\mathcal{A}$'s challenge bit is that of $\mathcal{B}$'s, and thus $\Pr[\mathsf{S}|\mathsf{G}]$ is equal to the probability that $\mathcal{A}$ succeeds in guessing the challenge bit in the `CIPR` game. This implies $2|\Pr[\mathsf{S}|\mathsf{G}] - 1/2| = \mathtt{Adv}^{\mathtt{CIPR}}_{\mathcal{H},q,\mathcal{A},\Phi}(\lambda)$. On the other hand, if G does not occur, $\mathcal{B}$ uses a uniformly chosen random bit as its final output bit σ', which implies $\Pr[\mathsf{S}|\overline{\mathsf{G}}] = 1/2$. Using the above in Equation (14), we obtain Equation (13), as required. □ (**Lemma 3**)

Theorem 6 follows from the combination of Lemmas 2 and 3. □ (**Theorem 6**)

6.3 Bounded RKA-Secure PRF

Finally, we show that by combining a (q,Φ)-`CIPR`-secure function family with a standard PRF, we obtain a PRF that provides Φ-RKA security, as long as an adversary uses at most q functions for deriving related keys in the security game. We stress that although the number of *functions* is a-priori bounded by q, the number of *evaluations* that an adversary may observe (through Eval queries) is unbounded. We refer to this slightly weaker variant of Φ-RKA security of a PRF as (q,Φ)-`RKA` *security*.

We formally define (q,Φ)-`RKA` security of a PRF via the security game shown in Figure 6. This game is a simple modification of the PRF game in Section 2.4. Specifically, in the (q,Φ)-`RKA` security game, an initial key k^* is picked, and the game maintains a counter *ctr* (initialized to 0) that tracks the number of related keys the adversary has requested. The oracle RKD (which stands for **R**elated-**K**ey **D**erivation) takes a function $\phi \in \Phi$ as input, increments the counter

$\underline{\mathrm{RKAPRF}^{\mathtt{F}}_{q,\mathcal{A},\Phi}(\lambda)}$:
$par \leftarrow \mathtt{F.Setup}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$
$\mathcal{F} \leftarrow \emptyset$
$k^* \leftarrow \mathtt{F.KeyGen}(par)$
$ctr \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\text{FUNC,RKD}}(par)$
return $(b = b')$

$\underline{\text{proc. EVAL}(i, x)}$:
if $i > ctr$, return $\perp$
if $b = 1$
  $y \leftarrow \mathtt{F.Eval}(k_i, x)$
else
  if $\mathcal{F}[i, x] = \perp$, $\mathcal{F}[i, x] \leftarrow_\$ R$
  $y \leftarrow \mathcal{F}[i, x]$
return y

$\underline{\text{proc. RKD}(\phi \in \Phi_\lambda)}$:
If $ctr > q$,
  return $\perp$
$ctr \leftarrow ctr + 1$
$k_{ctr} \leftarrow \phi(k^*)$
return ctr

Fig. 6. Game defining (q, Φ)-RKA security of a PRF.

Alg. $\widehat{\mathtt{F}}.\mathtt{Setup}(1^\lambda)$:
$par' \leftarrow \mathtt{F.Setup}(1^\lambda)$
$H \leftarrow_\$ \mathcal{H}_\lambda$
$par \leftarrow (par', H)$
return par

Alg. $\widehat{\mathtt{F}}.\mathtt{KeyGen}(par)$:
$k \leftarrow_\$ D_\lambda$
return k

Alg. $\widehat{\mathtt{F}}.\mathtt{Eval}(par, k, x)$:
$(par', H) \leftarrow par$
$\widetilde{k} \leftarrow H(k)$
$y \leftarrow \mathtt{F.Eval}(par', \widetilde{k}, x)$
return y

Fig. 7. (q, Φ)-RKA-secure PRF $\widehat{\mathtt{F}}$ constructed from a standard PRF F and a (q, Φ)-CIPR-secure function family $\mathcal{H}$.

$ctr \leftarrow ctr + 1$, computes a related key $k_{ctr} \leftarrow \phi(k^*)$, and returns the handle ctr that can be used in an EVAL query to specify the index of the key under which an adversary wish to see an evaluation result. Furthermore, like the HASH oracle in the CIPR game, the oracle RKD can be used at most q times, and all functions used in RKD queries are required to be distinct. However, we again stress that there is no restriction on the number of queries on the EVAL oracle.

Definition 14 *Let Φ be a function family, and let the advantage of an adversary $\mathcal{A}$ playing the security game in Figure 6 with respect to a PRF* $\mathtt{F} = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Eval})$ *be defined as*

$$\mathtt{Adv}^{\mathtt{RKAPRF}}_{\mathtt{F},q,\mathcal{A},\Phi}(\lambda) = 2\left|\Pr[\mathrm{RKAPRF}^{\mathtt{F}}_{q,\mathcal{A},\Phi}(\lambda) \Rightarrow 1] - \frac{1}{2}\right|.$$

F *is said to be a* (q, Φ)-RKA *secure if for all PPT adversaries* $\mathcal{A}$, $\mathtt{Adv}^{\mathtt{RKAPRF}}_{\mathtt{F},q,\mathcal{A},\Phi}(\lambda)$ *is negligible in the security parameter* λ.

We will now show how we construct a (q, Φ)-RKA secure PRF. Let $\mathcal{H} = \{H : D \to R\}$ be a family of functions with domain $D = D_\lambda$ and range $R = R_\lambda$, and let F be a PRF. We assume that the key space of F is R, and furthermore that $\mathtt{F.KeyGen}(par)$ just samples a uniformly random element from R, and outputs this as a key, for any par output from $\mathtt{F.Setup}(1^\lambda)$. Using these components, we construct another pseudorandom function $\widehat{\mathtt{F}}$ as in Figure 7. Note that the key space of $\widehat{\mathtt{F}}$ (when set up with the security parameter λ) is D (which is equal to the domain of the hash function $H \in \mathcal{H}$).

Theorem 7 *Let $q = q(\lambda)$ be any positive polynomial, let $\Phi = \{\phi : D \to D\}$ be a family of functions with domain and range $D = D_\lambda$, and let $\mathcal{H} = \{H : D \to R\}$*

Alg. $\texttt{PRF-PKE.Setup}(1^\lambda)$:
$par' \leftarrow \texttt{PKE.Setup}(1^\lambda)$
$par'' \leftarrow \widehat{\texttt{F}}.\texttt{Setup}$
$par \leftarrow (par', par'')$
return par

Alg. $\texttt{PRF-PKE.KeyGen}(par)$:
$(par', par'') \leftarrow par$
$(pk, sk) \leftarrow \texttt{PKE.KeyGen}(par')$
return (pk, sk)

Alg. $\texttt{PRF-PKE.Enc}(pk, m)$:
$r \leftarrow_\$ \mathcal{R}$
$\widetilde{r} \leftarrow \widehat{\texttt{F}}.\texttt{Eval}(par, r, pk||m)$
$c \leftarrow \texttt{PKE.Enc}(pk, m; \widetilde{r})$
return c

Alg. $\texttt{PRF-PKE.Dec}(sk, c)$:
$m \leftarrow \texttt{PKE.Dec}(sk, c)$
return m

Fig. 8. Scheme PRF-PKE constructed from a PKE scheme PKE and a PRF $\widehat{\texttt{F}}$.

be a (q, Φ)-CIPR *secure family of (hash) functions with domain* D *and range* $R = R_\lambda$.[8] *Let* F *be a secure PRF with key space* R *(when set up with the security parameter* λ*), and with a key generation algorithm that outputs a uniformly random element from* R*. Then, the construction* $\widehat{\texttt{F}}$ *shown in Figure 7 is* (q, Φ)-RKA *secure. More precisely, for all PPT adversaries* $\mathcal{A}$*, there exist PPT adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$*, such that*

$$\texttt{Adv}^{\texttt{RKAPRF}}_{\widehat{\texttt{F}},q,\mathcal{A},\Phi}(\lambda) \leq \texttt{Adv}^{\texttt{CIPR}}_{\mathcal{H},q,\mathcal{B}_1,\Phi}(\lambda) + \texttt{Adv}^{\texttt{PRF}}_{\texttt{F},\mathcal{B}_2}(\lambda). \tag{15}$$

The intuition behind the proof of this theorem is fairly simple. Recall that (q, Φ)-CIPR security of the underlying hash family $\mathcal{H}$ essentially ensures the property that, for a randomly chosen function $H \leftarrow_\$ \mathcal{H}$, and for any functions $\phi_1, \ldots, \phi_q \in \Phi$, having access to the functions $\{\texttt{F.Eval}(H(\phi_i(k^*), \cdot)\}_{i\in[q]}$ is indistinguishable from having access to the functions $\{\texttt{F.Eval}(\widetilde{k}_i, \cdot)\}_{i\in[q]}$, where $k^* \in D$ and each $\widetilde{k}_i \in R$ are chosen uniformly at random. Then, the security of the PRF F ensures that the latter is indistinguishable from having access to q independently chosen random functions. The full proof of Theorem 7 can be found in the full version of the paper.

7 IND-RRR-CCA Security in the Standard Model

We will now show that, for any predetermined polynomial n, we can transform a PKE scheme PKE which is secure in the standard sense (without related-randomness security) into a scheme PRF-PKE that is (n, Φ, Ψ)-IND-RRR-CCA secure in the standard model, by using a (n, Θ)-RKA secure PRF for an appropriate function class Θ. This approach is similar to that of [43] and [34], but we obtain security for a much richer class of function families that captures non-algebraic functions, such as bit-flipping and bit-fixing functions.

[8] In this statement, the requirements regarding output-unpredictability and collision resistance on Φ are implicitly posed by the requirement that $\mathcal{H}$ is (q, Φ)-CIPR secure (c.f. Theorem 6 and Corollary 1).

More formally, the construction of PRF-PKE is as follows: let PKE be a PKE scheme for which the randomness space of PKE.Enc is $\{0,1\}^\lambda$, let $\widehat{\mathsf{F}}$ be a PRF with key space $\mathcal{R}$ and a key generation algorithm $\widehat{\mathsf{F}}.\mathtt{KeyGen}(par)$ returning a uniformly random element from $\mathcal{R}$ as a key, for any par output by $\widehat{\mathsf{F}}.\mathtt{Setup}(1^\lambda)$. Using these components, we construct a PKE scheme PRF-PKE as in Figure 8. Note that the randomness space of PRF-PKE.Enc is $\mathcal{R}$. The related-randomness security of PRF-PKE is guaranteed by the following theorem:

Theorem 8 *Let $n = n(\lambda)$ be an integer-valued positive polynomial. Let $\Phi = \{\phi : \mathcal{R} \to \mathcal{R}\}$ and $\Psi = \{\psi : \mathcal{R} \times \mathcal{R} \to \mathcal{R}\}$ be function families, where $\mathcal{R} = \mathcal{R}_\lambda$. Let Θ be the function family defined by using Φ and Ψ as follows:*

$$\Theta := \left\{ f(\cdot) := \phi(\psi(r,\cdot)) \middle| \phi \in \Phi, \psi \in \Psi, r \in \mathcal{R} \right\} \cup \Phi.$$

Let $\widehat{\mathsf{F}}$ be a (n,Θ)-RKA secure PRF[9], and let PKE be an IND-CCA secure PKE scheme. Then, the construction PRF-PKE shown in Figure 8 is (n,Φ,Ψ)-IND-RRR-CCA secure. More precisely, for all PPT (n,Φ,Ψ)-restricted adversaries $\mathcal{A}$ that make at most $q_r = q_r(\lambda)$ Refresh queries, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that

$$\mathtt{Adv}^{\mathtt{IND\text{-}RRR\text{-}CCA}}_{\mathtt{PRF\text{-}PKE},\mathcal{A}}(\lambda) \le 2(q_r+1)\mathtt{Adv}^{\mathtt{RKAPRF}}_{\widehat{\mathsf{F}},n,\mathcal{B}_1,\Theta}(\lambda) + \mathtt{Adv}^{\mathtt{IND\text{-}CCA}}_{\mathtt{PKE},\mathcal{B}_2}(\lambda). \tag{16}$$

The proof of Theorem 8 is based on a hybrid argument over the refresh epochs. More specifically, in each epoch, we use the (n,Θ)-RKA security of $\widehat{\mathsf{F}}$ to replace the output $\widetilde{r}$ with uniformly random values. This is possible since the randomness r' used in the response to LR and Enc oracle queries will correspond to related keys of $\widehat{\mathsf{F}}$ computed by $f \in \Theta$. More precisely, it will be either of the form $r' = \phi(r_1)$ (in the first epoch) or $r' = \phi(\psi_{j-1}(r_{j-1}, s_{j-1}))$ (in the $j(\ge 2)$-th epoch), where r_1 and s_{j-1} are chosen uniformly at random, and thus can be viewed as related keys of the initial key k^* in the RKA game by viewing r_1 or s_{j-1} as k^*. Note that the adversary is assumed to make in total at most n LR and Enc queries in each epoch, and thus (n,Θ)-RKA security will suffice. Then, in the last hybrid, the values $\widetilde{r}$ are all uniformly chosen, and we can rely on the IND-CCA security of the underlying PKE scheme PKE to conclude the proof. The full proof can be found in the full version of the paper.

Combining Theorem 8 with Corollary 1, we obtain the following corollary:

Corollary 2 *Let $t = t(\lambda)$, $p = p(\lambda)$, and $n = n(\lambda)$ be integer-valued positive polynomials such that t is always even and larger than 8. Let PKE be an IND-CCA secure PKE scheme, let F be a PRF, and let $\mathcal{H}$ be a t-wise independent hash family. Assume that the key space of F and the output space of $\mathcal{H}$ are $\{0,1\}^\lambda$ when F is set up with a security parameter λ. Let $\widehat{\mathsf{F}}$ be the PRF constructed from F and $\mathcal{H}$ as shown in Figure 6, and let PKE′ be the PKE scheme obtained from*

[9] In this statement, the requirements regarding output unpredictability and collision resistance of Φ and Ψ are implicitly implied as $\widehat{\mathsf{F}}$ is a (n,Θ)-RKA secure PRF (c.f. Theorem 7). This will be made explicit in Corollary 2.

$\mathtt{PKE}$ *and* $\widehat{\mathtt{F}}$ *as shown in Figure 8. Let* Φ *and* Ψ *be function families such that* $|\Phi| \leq 2^p$ *and* $|\Psi| \leq 2^{p'}$*, respectively. Assume that*

$$t \geq n(p + p' + \log|\mathcal{R}| + 2\lambda + 2), \tag{17}$$

$$\max\{\mathsf{UP}^{\Phi}(\lambda), \mathsf{sUP}^{\Phi,\Psi}(\lambda)\} \leq 2^{-(3n\lambda + O(\log\lambda))}, \tag{18}$$

$$\max\{\mathsf{CR}^{\Phi}(\lambda), \mathsf{sCR}^{\Phi,\Psi}(\lambda)\} \leq \binom{n}{2}^{-1} \cdot 2^{-n\lambda}. \tag{19}$$

Then, $\mathtt{PKE}'$ *is* (n, Φ, Ψ)-$\mathtt{IND\text{-}RRR\text{-}CCA}$ *secure. More precisely, for all PPT* (n, Φ, Ψ)*-restricted adversaries* $\mathcal{A}$ *that make at most* $q_r = q_r(\lambda)$ REFRESH *queries, there exist PPT adversaries* $\mathcal{B}$ *and* $\mathcal{B}'$ *such that*

$$\mathtt{Adv}^{\mathtt{IND\text{-}RRR\text{-}CCA}}_{\mathtt{PKE}',\mathcal{A}}(\lambda) \leq 12(q_r + 1) \cdot 2^{-\lambda} + 2(q_r + 1)\mathtt{Adv}^{\mathtt{PRF}}_{\mathtt{F},\mathcal{B}}(\lambda) + \mathtt{Adv}^{\mathtt{IND\text{-}CCA}}_{\mathtt{PKE},\mathcal{B}'}(\lambda).$$

Proof. (of Corollary 2) Note that each function $f \in \Theta$ can be specified by (1) a bit indicating whether f is in the set $\{\phi(\psi(r,\cdot))|\phi \in \Phi, \psi \in \Psi, r \in \mathcal{R}$ or in the set Φ, (2-1) a tuple $(\phi, \psi, r) \in \Phi \times \Psi \times \mathcal{R}$ in case f belongs to the former set, and (2-2) a function $\phi \in \Phi$ in case f belongs to the latter set. This implies that $|\Theta| \leq 2 \cdot (|\Phi| \cdot |\Psi| \cdot |\mathcal{R}| + |\Phi|) \leq 2^{p+p'+2} \cdot |\mathcal{R}| = 2^{p''}$. where $p'' = p + p' + 2 + \log|\mathcal{R}|$. Furthermore, by definition, the output unpredictability of Θ is at most the maximum of the output unpredictability of Φ and that of the seed-induced output-unpredictability of Φ with respect to Ψ, i.e. $\max\{\mathsf{UP}^{\Phi}(\lambda), \mathsf{sUP}^{\Phi,\Psi}(\lambda)\}$, and exactly the same relation holds for collision resistance. Recall also that the output space of $\mathcal{H}$ is $\{0,1\}^{\lambda}$.

Now, by using the definition of the function class Θ with the parameters described above in Corollary 1, we obtain the requirements in Equations (17), (18), and (19), and the upperbound $6 \cdot 2^{-\lambda}$ for the advantage of any (even computationally unbounded) adversary that attacks the (n, Θ)-$\mathtt{CIPR}$ security of $\mathcal{H}$. Then, using it in turn in Theorem 8, we obtain this corollary. □ (**Corollary 2**)

The reason the impossibility result from Section 4 is not applicable to the above construction, is that for each security parameter λ, with high probability over the choice of the t-wise independent hash function H, the function families Φ and Ψ are not capable of expressing H and thereby the encryption function of the scheme, due to requirement on t in Equation (17). Note also that, as the size of the description of H must be linear in t, and this description is part of the parameters par'' in Figure 8, the size of the parameters of the construction will grow linearly in the right hand side of Equation (17) i.e. linearly in number of queries an adversary is allowed to make in an epoch, and logarithmically in the size of the function families Φ and Ψ.

Acknowledgement. A part of this work was supported by JST CREST grant number JPMJCR1688. Jacob Schuldt was supported by JSPS KAKENHI Grant Number 15K16006.

References

1. M. Abdalla, F. Benhamouda, and A. Passelègue. An algebraic framework for pseudorandom functions and applications to related-key security. In *CRYPTO (1) 2015*, pp. 388–409, 2015.

2. M. Abdalla, F. Benhamouda and A. Passelgue. Algebraic XOR-RKA-Secure Pseudorandom Functions from Post-Zeroizing Multilinear Maps. Cryptology ePrint Archive, Report 2017/500, 2017.
3. M. Abdalla, F. Benhamouda, A. Passelègue, and K.G. Paterson. Related-key security for pseudorandom functions beyond the linear barrier. In *CRYPTO (1) 2014*, pp. 77–94, 2014.
4. B. Applebaum and E. Widder. Related-Key Secure Pseudorandom Functions: The Case of Additive Attacks. Cryptology ePrint Archive, Report 2014/478, 2014.
5. M.R. Albrecht, P. Farshim, K.G. Paterson and G.J. Watson On cipher-dependent related-Key attacks in the ideal-cipher model. In *FSE 2011*, pp. 128–145, 2011.
6. M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In *ASIACRYPT 2009*, pp. 232–249, 2009.
7. M. Bellare and D. Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO 2010*, pp. 666–684, 2010.
8. M. Bellare, D. Cash, and R. Miller. Cryptography secure against related-key attacks and tampering. In *ASIACRYPT 2011*, pp. 486–503, 2011.
9. M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In *EUROCRYPT 2003*, pp. 491–506, 2003.
10. M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *FOCS 1994*, pp. 276–287, 1994.
11. M. Bellare and B. Tackmann. Nonce-based cryptography: Retaining security when randomness fails. In *EUROCRYPT (1) 2016*, pp. 729–757, 2016.
12. M. Bendel. Hackers describe PS3 security as epic fail, gain unrestricted access. 2011. http://www.exophase.com/20540/hackers-describe-ps3-security-as-epic-fail-gain-unrestricted-access/.
13. D.J. Bernstein, Y.-A. Chang, C.-M. Cheng, L.-P. Chou, N. Heninger, T. Lange, and N. van Someren. Factoring RSA keys from certified smart cards: Coppersmith in the wild. Cryptology ePrint Archive, Report 2013/599, 2013.
14. E. Birrell, K.-M. Chung, R. Pass, and S. Telang. Randomness-dependent message security. In *TCC 2013*, pp. 700–720, 2013.
15. M. Bellare, R. Canetti, and H. Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *FOCS 1996*, pp. 514–523, 1996.
16. Bitcoin.org. Android security vulnerability. 2013. http://bitcoin.org/en/alert/2013-08-11-android.
17. Y. Chen, B. Qin, J. Zhang, Yi Deng, and S.S.M. Chow. Non-malleable functions and their applications. In *PKC (2) 2016*, pp. 386–416, 2016.
18. I. Damgård, S. Faust, P. Mukherjee, and D. Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. In *ASIACRYPT (2) 2013*, pp. 140–160, 2013.
19. I. Damgård, S. Faust, P. Mukherjee, and D. Venturi. The chaining .emma and its application. In *ICITS 2015*, pp. 181–196, 2015.
20. Debian. Debian Security Advisory DSA-1571-1: OpenSSL – predictable random number generator. 2008. http://www.debian.org/security/2008/dsa-1571.
21. Y. Dodis. Exposure-resilient cryptography. PhD thesis, Massachusetts Institute of Technology, 2000.
22. Y. Dodis and Y. Yu. Overcoming weak expectations. In *TCC 2013*, pp. 1–22, 2013.
23. L. Dorrendorf, Z. Gutterman, and B. Pinkas. Cryptanalysis of the random number generator of the Windows operating system. *ACM Trans. Inf. Syst. Secur.*, 13(1), 2009.

24. I. Goldberg and D. Wagner. Randomness and the Netscape browser. 1996. http://www.drdobbs.com/windows/184409807.
25. V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. In *TCC 2011*, pp. 182–200, 2011.
26. Z. Gutterman and D. Malkhi. Hold your sessions: An attack on java session-id generation. In *CT-RSA 2005*, pp. 44–57, 2005.
27. Z. Gutterman, B. Pinkas, and T. Reinman. Analysis of the linux random number generator. In *S & P*, pp. 371–385, 2006.
28. J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. Construction of a pseudorandom generator from any one-way function. *SIAM J. Computing*, 28(4):1364–1396, 1999.
29. N. Heninger, Z. Durumeric, E. Wustrow, and J.A. Halderman. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *USENIX Security 2012*, pp. 205–220, 2012.
30. V.T. Hoang, J. Katz, A. O'Neill, and M. Zaheri. Selective-opening security in the presence of randomness failures. In *ASIACRYPT (2) 2016*, pp. 278–306, 2016.
31. S. Kamara and J. Katz How to encrypt with a malicious random number generator. In *FSE 2008*, pp. 303–315, 2008.
32. A.K. Lenstra, J.P. Hughes, M. Augier, J.W. Bos, T. Kleinjung, and C. Wachter. Public keys. In *CRYPTO 2012*, pp. 626–642, 2012.
33. National Institute of Standards and Technology (NIST). Fips publication 186: Digital signature standards (dss). 1994.
34. K.G. Paterson, J.C.N. Schuldt, and D.L. Sibborn. Related randomness attacks for public key encryption. In *PKC 2014*, pp. 465–482, 2014.
35. K.G. Paterson, J.C.N. Schuldt, D.L. Sibborn, and H. Wee. Security against related randomness attacks via reconstructive extractors. In *IMA Cryptography and Coding 2015*, pp. 23–40, 2015.
36. T. Pornin RFC6979: Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA), 2013. https://tools.ietf.org/html/rfc6979
37. B. Qin, S. Liu, T.H. Yuen, R.H. Deng, and K. Chen. Continuous non-malleable key derivation and its application to related-key security. In *PKC 2015*, pp. 557–578, 2015.
38. A. Raghunathan, G. Segev, and S.P. Vadhan. Deterministic public-key encryption for adaptively chosen plaintext distributions. In *EUROCRYPT 2013*, pp. 93–110, 2013.
39. T. Ristenpart and S. Yilek. When good randomness goes bad: Virtual machine reset vulnerabilities and hedging deployed cryptography. In *NDSS 2010*, 2010.
40. P. Rogaway and T. Shrimpton. A provable-security treatment of the key-wrap problem. In *EUROCRYPT 2006*, pp. 373–390, 2006.
41. L. Trevisan and S.P. Vadhan. Extracting randomness from samplable distributions. In *FOCS 2000*, pp. 32–42, 2000.
42. D. Vergnaud and D. Xiao. Public-Key Encryption with Weak Randomness: Security against Strong Chosen Distribution Attacks. Cryptology ePrint Archive, Report 2013/681, 2013.
43. S. Yilek. Resettable public-key encryption: How to encrypt on a virtual machine. In *CT-RSA 2010*, pp. 41–56, 2010.